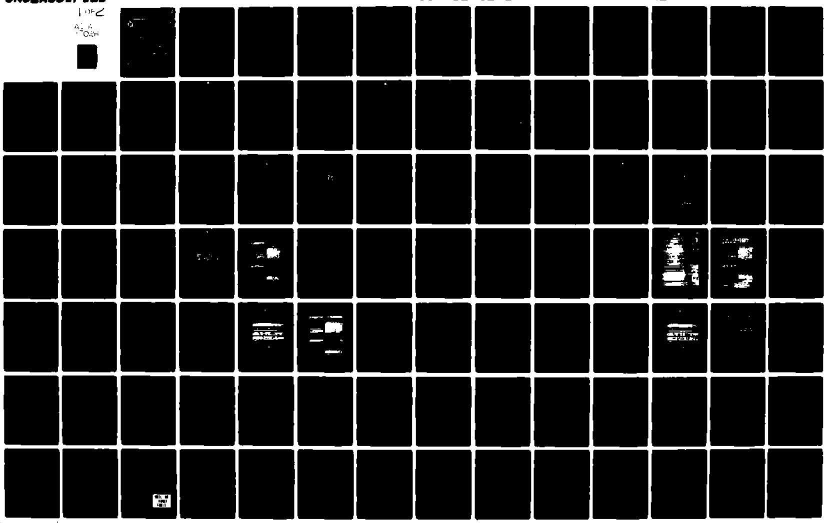
AD-A117 026 DELTA INFORMATION SYSTEMS INC JENKINTOWN PA F/G 14/3
GRAPHIC/SYMBOL SEGMENTATION FOR GROUP 4 FACSIMILE SYSTEMS.(U)
APR 82 A R DEUTERMANN DCA100-81-C-0024
UNCLASSIFIED NCS-TIB-82-2 NL

1 OF 2



(2)

NCS TIB 82-2

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 82-2

GRAPHIC/SYMBOL SEGMENTATION FOR GROUP 4 FACSIMILE SYSTEMS

APRIL 1982

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

DISCLAIMER NOTICE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NCS TIB 82-2	2. GOVT ACCESSION NO. AD-A119026	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Graphic/Symbol Segmentation for Group 4 Facsimile Systems		5. TYPE OF REPORT & PERIOD COVERED FINAL 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Alan R. Deutermann		8. CONTRACT OR GRANT NUMBER(s) DCA100-81-C-0024
9. PERFORMING ORGANIZATION NAME AND ADDRESS Delta Information Systems, Inc. 259 Wyncote Road Jenkintown, PA 19046		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Communications System Office of Technology and Standards Washington, DC 20305		12. REPORT DATE APRIL 1, 1982 13. NUMBER OF PAGES 114
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Facsimile
Group 4
Graphic
Symbol
Graphic/Symbol Segmentation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this study was to examine possible techniques for segmenting a document into graphic and symbol areas, and assemble a code that represents the entire document. Parameters to be considered include compression, commonality with facsimile and TELETEX* transmissions, and complexity of implementation. Six segmentation technique were selected for analysis. The techniques were designed to differ from each other as much as possible, so as to display a wide variety of characteristics. For each technique, many minor modifications

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

would be possible, but it is not expected that these modifications would alter the conclusions drawn from the study.

The segmentation techniques analyzed are:

- SYMBOL REMOVAL/SCAN LINE
- SYMBOL REMOVAL/FULL DOCUMENT
- SYMBOL REMOVAL/LINE OF SYMBOLS
- EXTENDED TELETEX
- PARTIAL LINE OF SYMBOLS
- FULL LINE OF SYMBOLS

NCS TECHNICAL INFORMATION BULLETIN 82-2

GRAPHIC/SYMBOL SEGMENTATION

FOR

GROUP 4 FACSIMILE SYSTEMS

APRIL 1982

PROJECT OFFICER	APPROVED FOR PUBLICATION:
	Marshall L. Cain
DENNIS BODSON Senior Electronics Engineer Office of NCS Technology and Standards	MARSHALL L. CAIN Assistant Manager Office of Technology and Standards

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunication Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the Electronic Industries Association, the American National Standards institute, the International Organization for Standardization, and the International Telegraph and Telephone Consultative Committee of the International Telecommunication Union. This Technical Information Bulletin presents an overview of an effort which is contributing to the development of compatible Federal, national, and international standards in the area of digital facsimile standards. It has been prepared to inform interested Federal activities of the progress of these efforts. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS
Washington, D.C. 20305
(202) 692-2124

GRAPHIC/SYMBOL SEGMENTATION

for Group 4 Facsimile Systems

April 1, 1982

Final Report

Submitted To:

NATIONAL COMMUNICATIONS SYSTEM

Office of Technology and Standards
Washington, D.C. 20305

Contracting Agency:

DEFENSE COMMUNICATIONS AGENCY

Contract Number

DCA100-81-C-0024

Submitted by:

D E L T A I N F O R M A T I O N S Y S T E M S, I N C.

310 Cottman Street

Jenkintown, Pa. 19046

CONTENTS

Appendix A International Digital Facsimile Coding Standards

Apendix B Combined Symbol Matching Facsimile Data Compression System

Appendix C Proposal for Mixed Character Coded Text and Facsimile Areas on the Same Page

1. INTRODUCTION

This report summarizes the results of work performed under contract DCA100-81-C-0024 with the Defense Communications Agency. Consideration is now being given to possible CCITT standards for Group 4 Facsimile which refers to the transmission of an A4 sized page over data networks containing error control. It is likely that the basic coding technique for Group 4 transmissions will be some advanced form of the Modified READ code,which is the optional compression algorithm for Group 3. The purpose of this study is to investigate the more advanced mixed mode coding techniques for possible consideration as an optional algorithm for Group 4. In a mixed mode system the information printed on a page is divided into two parts - symbols (letters, numerals, punctuation, etc.) and graphics (logos, signatures, sketches, etc.) The purpose of this study was to examine possible techniques for segmenting a document into graphic and symbol areas, and assemble a code that represents the entire document. Parameters to be considered include compression, commonality with facsimile and TELETEX* transmissions, and complexity of implementation.

Six segmentation technique were selected for analysis. The techniques were designed to differ from each other as

*TELETEX refers to a CCITT recommendation which is now under development for communication between word processors.

much as possible, so as to display a wide variety of characteristics. For each technique, many minor modifications would be possible, but it is not expected that these modifications would alter the conclusions drawn from the study.

The segmentation techniques analyzed are:

- SYMBOL REMOVAL/SCAN LINE
- SYMBOL REMOVAL/FULL DOCUMENT
- SYMBOL REMOVAL/LINE OF SYMBOLS
- EXTENDED TELETEX
- PARTIAL LINE OF SYMBOLS
- FULL LINE OF SYMBOLS

Section 2 presents the methodology used for measuring compression. Section 3 presents descriptions of the six mixed-mode segmentation alternatives considered, together with the calculations resulting in compression estimates. Section 4 discusses the commonality of each alternative with Group 3 facsimile, Group 4 facsimile, and with TELETEX. Section 5 discusses the complexity of implementation of each technique. Section 6 compares the alternatives and draws conclusions, while recommendations are made in Section 7.

2.0 METHODOLOGY FOR MEASURING COMPRESSION

For each of the six proposed mixed mode techniques, an estimate of the compression has been made. Where applicable, an additional estimate was made of the compression using a Carriage Return/Line Feed (CR/LF) symbol to terminate a line of symbols. Estimates of compression were made for CCITT test Documents 1 and 5. Document 5 was slightly modified by removing the vertical line in the center of the page. If this were not done, the technique FULL LINE OF SYMBOLS would not be able to encode any symbols.

It should be emphasized that the compression values calculated in this report are estimates only, and should not be regarded as actual measured numbers. However, it is expected that the relative compressions of the various segmentation techniques are accurate, since the same assumptions and estimates were used for all of them.

2.1 ASSUMPTIONS

In making compression estimates the following assumptions were made:

(1) Each symbol is encoded using 8 bits, which allows up to 256 different symbols.

(2) Several of the 256 symbol codes can be made available for indicating the termination of symbol transmission, or other requirements of the segmentation technique employed.

(3) A stored library, suitable for the document being

transmitted, is available at both sending and receiving terminals.

(4) Bits required to identify the proper symbol library to the receiving terminal are neglected.

(5) The stored library will accommodate either fixed or proportionaly spaced fonts, including several widths for word spaces.

(6) All characters of the principal font used in the document are in fact recognized as such, and will be encoded as symbols, subject to the rules of the proposed technique.

(7) Lines of symbols can be accumulated despite slight skews of the printed lines.

(8) The characters of the principal font include math symbols, italics, and Greek letters, but not subscripts or superscripts, or long horizontal or vertical lines.

(9) Graphic data is transmitted using the modified READ code, without EOL's.

(10) The number of bits required to transmit increased width of white spaces by means of Modified READ can be neglected. This is because the spacing between groups of black pels (such as symbols) usually only has to be specified once, and the READ code length does not grow rapidly with the length of a white run.

(11) Each document consists of 2,376 rows with 1,728 pels

per row (7.7 pels per mm, or approximately 196 pels per inch). See Appendix A.

(12) Code transmissions will not experience any transmission errors, so addition of redundancy for error control is not required.

2.2 PRELIMINARY CALCULATIONS

2.2.1 BITS PER SYMBOL USING MODIFIED READ CODE

It is assumed that there is an average number of bits required to transmit a text symbol by means of the Modified READ code. In order to obtain this average, CCITT Document 4 (see figure 2-1) was used, which is almost 100% text. Document 4 was found to have a total of 4,001 symbols. Appendix A indicates that it took 585,074 bits to transmit this document using the Modified READ code, with $K = \infty$. Included in this total are 2,376 EOL's at 12 bits each or 28,512 bits, and about 454 blank scan lines (not including those between adjacent lines of text), each of which takes one bit to code, or 454 bits. Subtracting both of these from the total, it took 556,108 bits to transmit the text, or $\frac{556{,}108}{4001}$ = 138.992 bits per symbol.

2.2.2 GRAPHIC BITS PER DOCUMENT

2.2.2.1 DOCUMENT 1

Now consider Document 1, shown in Figure 2-2. Appendix A indicates that it took 175,704 bits to transmit with Modified READ ($K = \infty$), which less 28,512 bits for EOL's, gives 147,192 bits for the information on the page. It is

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.

Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet,et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.

A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.

Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.

Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.

L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.

L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération, très puissants et dotés de volumineuses mémoires de masse, a conduit à en réduire substantiellement le nombre.

L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger,et posant des problèmes délicats de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre "gérera" environ un million d'abonnés à la fin du VIème Plan.

La mise en place de ces centres a débuté au début de l'année 1971 : un ordinateur IRIS 50 de la Compagnie Internationale pour l'Informatique a été installé à Toulouse en février ; la même machine vient d'être mise en service au centre de calcul interrégional de Bordeaux.

Figure 2-1 CCITT Document 4

Photo n° 1 - Document très dense lettre 1,5mm de haut -

 Restitution photo n° 9

THE SLEREXE COMPANY LIMITED

SAPORS LANE · BOOLE · DORSET · BH 25 8 ER

TELEPHONE BOOLE (945 13) 51617 · TELEX 123456

Our Ref. 350/PJC/EAC   18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Figure 2-2 CCITT Document 1

Registered in England: No. 2038
Registered Office: 60 Vicars Lane, Ilford. Essex.

assumed that the Combined Symbol Matching (CSM) Technique (Appendix B), which uses a transmitted library, encodes all characters on the page, with the exception of the "SLEREXE" line, whose characters are too large for the symbol block size. By actual count, the number of symbols that would be coded is 937, not including spaces. (In fact, Appendix B shows that 988 symbols were recognized.) Using Modified READ, this should take 937 x 138.922 = 130,236 bits. Therefore the number of bits required to transmit the graphics on Document 1 is 147,192 - 130,236 = 16,956 bits. This compares with the 18,549 bits used to transmit Residue Code given by Appendix B. An alternate calculation of the average bits per symbol can be obtained by subtracting the 18,549 residue bits of Appendix B from the 147,192 bits to transmit the whole document, giving 128,643 bits to transmit the text symbols. Dividing this by the 937 symbols gives an average of 137.29 bits per symbol, which compares favorably with 138.992 obtained in Section 2.2.1.

In a stored library approach, it is assumed that the letterhead lines, except the "SAPORS LANE" line, would not be encoded because the character sizes differ greatly from the characters in the body of the text. This means that 107 symbols out of the 937 would not be encoded, which would take 107 x 138.992 = 14,872 bits to transmit by Modified READ. Therefore the total residue that would not be symbol encoded would be 16,956 + 14,872 = 31,828 bits for Document 1.

L'ordre de lancement et de réalisation des applications fait l'objet de décisions au plus haut niveau de la Direction Générale des Télécommunications. Il n'est certes pas question de construire ce système intégré "en bloc" mais bien au contraire de procéder par étapes, par paliers successifs. Certaines applications, dont la rentabilité ne pourra être assurée, ne seront pas entreprises. Actuellement, sur trente applications qui ont pu être globalement définies, six en sont au stade de l'exploitation, six autres se sont vu donner la priorité pour leur réalisation.
Chaque application est confiée à un "chef de projet", responsable successivement de sa conception, de son analyse-programmation et de sa mise en oeuvre dans une région-pilote. La généralisation ultérieure de l'application réalisée dans cette région-pilote dépend des résultats obtenus et fait l'objet d'une décision de la Direction Générale. Néanmoins, le chef de projet doit dès le départ considérer que son activité a une vocation nationale donc refuser tout particularisme régional. Il est aidé d'une équipe d'analystes-programmeurs et entouré d'un "groupe de conception" chargé de rédiger le document de "définition des objectifs globaux" puis le "cahier des charges" de l'application, qui sont adressés pour avis à tous les services utilisateurs potentiels et aux chefs de projet des autres applications. Le groupe de conception comprend 6 à 10 personnes représentant les services les plus divers concernés par le projet, et comporte obligatoirement un bon analyste attaché à l'application.

II - L'IMPLANTATION GEOGRAPHIQUE D'UN RESEAU INFORMATIQUE PERFORMANT

L'organisation de l'entreprise française des télécommunications repose sur l'existence de 20 régions. Des calculateurs ont été implantés dans le passé au moins dans toutes les plus importantes. On trouve ainsi des machines Bull Gamma 30 à Lyon et Marseille, des GE 425 à Lille, Bordeaux, Toulouse et Montpellier, un GE 437 à Massy, enfin quelques machines Bull 300 TI à programmes câblés étaient récemment ou sont encore en service dans les régions de Nancy, Nantes, Limoges, Poitiers et Rouen ; ce parc est essentiellement utilisé pour la comptabilité téléphonique.
A l'avenir, si la plupart des fichiers nécessaires aux applications décrites plus haut peuvent être gérés en temps différé, un certain nombre d'entre eux devront nécessairement être accessibles, voire mis à jour en temps réel : parmi ces derniers le fichier commercial des abonnés, le fichier des renseignements, le fichier des circuits, le fichier technique des abonnés contiendront des quantités considérables d'informations.
Le volume total de caractères à gérer en phase finale sur un ordinateur ayant en charge quelques 500 000 abonnés a été estimé à un milliard de caractères au moins. Au moins le tiers des données seront concernées par des traitements en temps réel.
Aucun des calculateurs énumérés plus haut ne permettait d'envisager de tels traitements.
L'intégration progressive de toutes les applications suppose la création d'un support commun pour toutes les informations, une véritable "Banque de données", répartie sur des moyens de traitement nationaux et régionaux, et qui devra rester alimentée, mise à jour en permanence, à partir de la base de l'entreprise, c'est-à-dire les chantiers, les magasins, les guichets des services d'abonnement, les services de personnel etc.
L'étude des différents fichiers à constituer a donc permis de définir les principales caractéristiques du réseau d'ordinateurs nouveaux à mettre en place pour aborder la réalisation du système informatif. L'obligation de faire appel à des ordinateurs de troisième génération très puissants et dotés de volumineuses mémoires de masse a conduit à en réduire substantiellement le nombre.
L'implantation de sept centres de calcul interrégionaux constituera un compromis entre : d'une part le désir de réduire le coût économique de l'ensemble, de faciliter la coordination des équipes d'informaticiens; et d'autre part le refus de créer des centres trop importants difficiles à gérer et à diriger, et posant des problèmes ardus de sécurité. Le regroupement des traitements relatifs à plusieurs régions sur chacun de ces sept centres permettra de leur donner une taille relativement homogène. Chaque centre gérera environ un million d'abonnés à la fin du VIème Plan.
La mise en place de ces centres a débuté au début de l'année 1971 : un ordinateur IRIS 50 de la Compagnie Internationale pour l'Informatique a été installé à Toulouse en février ; la même machine vient d'être mise en service au centre de calcul interrégional de Bordeaux.

Photo n° 1 - Document très dense lettre 1,5mm de haut -
Restitution photo n° 9

Figure 2-1 CCITT Document 4

THE SLEREXE COMPANY LIMITED

SAPORS LANE . BOOLE . DORSET . BH 25 8 ER

TELEPHONE BOOLE (945 13) 51617 . TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Figure 2-2 CCITT Document 1

2.2.2.2 DOCUMENT 5

Now consider Document 5, as shown in Figure 2-3. Appendix A gives 288,655 bits for Modified READ ($K = \infty$), less 28,512 bits for EOL's, gives 260,143 bits for the information on the page. An actual count of encoded symbols (including math symbols and Greek letters but not subscripts or superscripts) is 1,599. This should take 1,599 x 138.992 = 222,249 bits, which subtracted from 260,143 bits for the entire image, gives 37,894 bits for graphics. This compares with the 42,014 residue bits given by Appendix B for CSM. Subtracting the CSM residue from 260,143 bits gives 218,129 bits for transmitting the symbols by Modified READ, or an average of $\frac{218,129}{1,599}$ = 136.42 bits per symbol, which again compares favorably with the 138.992 bits per symbol from section 2.2.1.

Therefore, the bits required to transmit the information which cannot be symbol encoded is estimated to be 37,894 for Document 5. To this must be added bits for symbols that cannot be encoded using a particular technique.

2.2.3 SCAN LINES WITH SYMBOL DETECTIONS

The basis for most of the techniques is the detection of symbols as the scan lines touch the top of the symbol. The number of scan lines on which symbols are detected is obtained from Appendix B. This is given by the bits for COLADD divided by 11. For Document 1, this is $\frac{2,497}{11}$ = 227, or about 8.4 scan lines per line of text. Since 3 lines of text that were encoded by CSM will not be encoded in the proposed techniques, 25 scan lines are subtracted, giving 202 scan lines with symbol detections for Document 1. For

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente.

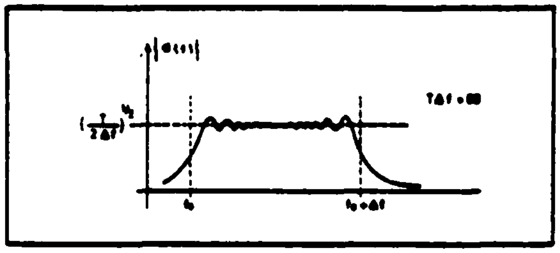


Fig. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leq f \leq f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composants le traversant ;

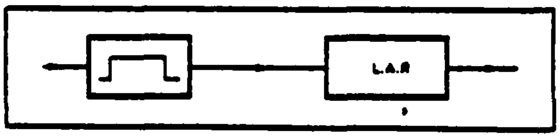


Fig. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_R décroissant linéairement avec la fréquence f suivant l'expression :

$$T_R = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).

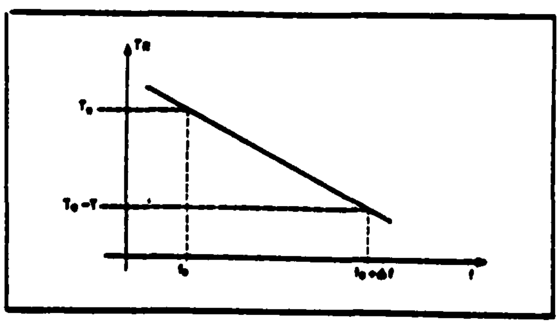


Fig. 4

telle ligne à retard est donnée par :

$$\varphi = -2\pi\int_0^f T_R\,\mathrm{d}f$$

$$\varphi = -2\pi\left[T_0+\frac{f_0 T}{\Delta f}\right]f+\pi\frac{T}{\Delta f}f^2$$

Et cette phase est bien l'opposé de $\underline{/\phi(f)}$,

à un déphasage constant près (sans importance) et à un retard T_0 près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0+\Delta f$, et nulle de part et d'autre de f_0 et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\dfrac{T}{1/\Delta f} = T\Delta f$

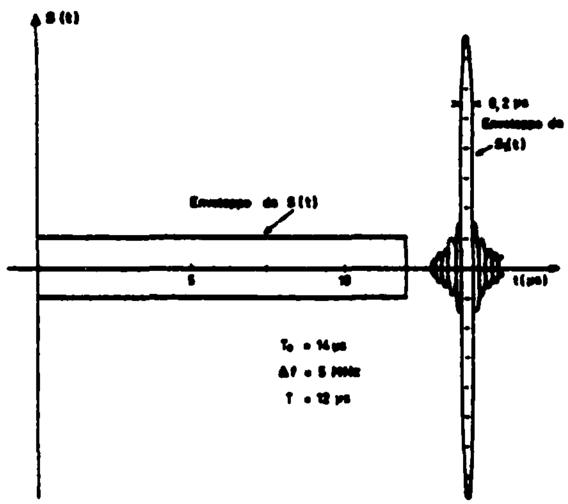


Fig. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0, qui met un temps T_0 pour traverser. La fréquence f entre à l'instant $t = (f-f_0)\dfrac{T}{\Delta f}$ et elle met un temps $T_0-(f-f_0)\dfrac{T}{\Delta f}$ pour traverser, ce qui la fait ressortir à l'instant T_0 également. Ainsi donc, le signal $S(t)$

Document 5, $\frac{4,895}{11}$ = 445 scan lines will have symbol detections.

2.3 INCREASED RESOLUTION

Consideration is being given to having the option of higher resolution for Group 4 Facsimile. Therefore it was desired to calculate the expected compression that could be obtained using 300 pels per inch and 400 pels per inch. Data comparable to that used for estimating compression for 200 pels per inch was not available.

It was felt that the number of bits required to transmit a document with Modified READ code is approximately <u>linearly</u> proportional to the resolution in pels per inch, despite the fact that the total number of pels in the document is proportional to the <u>square</u> of the resolution. The increase in bits is primarily due to the increased number of scan lines. The increased width of the scan lines does not greatly increase the number of bits required, and the finer spacing of the scan lines tends to produce more efficient vertical coding, since a scan line looks more like the previous scan line than with lower resolution. To confirm this impression, Group 4 Modified READ data was obtained for 200, 300 and 400 pels per inch for Document 1 and Document 5. The number of bits required to transmit these documents is shown in Table 2-1. This data is not comparable to that of Appendix A for 200 pels per inch because a different scan was used, which resulted in fewer bits required. In addition, Table 2-1 shows the number of bits required for each resolution divided by the number of

RESOLUTION lpi	RESOLUTION / 200	DOCUMENT 1		DOCUMENT 5	
		Bits/Page	Bits/Bits for 200 lpi	Bits/Page	Bits/Bits for 200 lpi
200	1.0	132,034	1.000	229,204	1.000
300	1.5	225,499	1.708	350,538	1.529
400	2.0	272,312	2.062	468,005	2.042

Table 2-1

COMPRESSED BITS/PAGE FOR VARIOUS RESOLUTIONS

bits required for 200 pels per inch. From these ratios, it can be seen that in most cases the assumption that bits increase linearly with resolution is a good one.

Table 2-2 shows the basic constants used in making the estimates of compression. The values for 200 lpi are obtained from Sections 2.1 and 2.2. For the other resolutions, Pels per Scan Line, Scan Lines, Bits per Symbol in READ Mode, and Graphics Bits per Document are all increased in direct proportion to the resolution. Scan Lines with Symbol Detections also increases in proportion to resolution, but since it cannot exceed the number of symbols on the document, there is obviously a limit beyond which proportionality will not apply. Therefore the values are decreased slightly below proportionality, especially for 400 lpi. Bits required for Horizontal and Vertical Position is adjusted to account for the number of pels per scan line and number of scan lines respectively.

The numbers from Table 2-2 are then used to calculate compression for 300 and 400 lpi in the same way that the compression for 200 lpi is calculated.

2.4 CALCULATING COMPRESSION

The number of bits required to construct the message is totaled. This includes symbol codes, graphics, mode changes, and horizontal and vertical positions. The compression is calculated by dividing the total message bits into the total number of image pels, which is always 2,376 x 1,728 = 4,105,728.

RESOLUTUION lpi	PELS PER SCAN LINE	SCAN LINES	BITS PER SYMBOL IN READ MODE	GRAPHICS BITS PER DOCUMENT		SCAN LINES WITH SYMBOL DETECTION		BITS REQUIRED FOR	
				DOCUMENT 1	DOCUMENT 5	DOC 1	DOC 5	HORIZONTAL POSITION OR WIDTH	VERTICAL POSITION
200	1,728	2,376	138.992	31,828	37,894	202	445	11	12
300	2,592	3,564	208.5	47,742	56,841	303	667	12	12
400	3,456	4,752	278.0	63,656	75,788	400	880	12	13

TABLE 2-2

CONSTANTS USED FOR COMPRESSION ESTIMATES AT VARIOUS RESOLUTIONS

3.0 MIXED MODE ALTERNATIVES

In this section each of the six mixed-mode segmentation techniques are considered in turn. The techniques are:

SYMBOL REMOVAL/SCAN LINE

SYMBOL REMOVAL/FULL DOCUMENT

SYMBOL REMOVAL/LINE OF SYMBOLS

EXTENDED TELETEX

PARTIAL LINE OF SYMBOLS

FULL LINE OF SYMBOLS

For each technique, a description is given of the approach and the compression for Documents 1 and 5 is estimated. The description of the approach includes a sketch of the composition of a portion of a mixed-mode transmission. In addition, images of Documents 1 and 5 are presented showing the areas transmitted by symbols (shaded), the lines of symbols constructed (enclosed by lines), and the area not transmitted because of the use of an optional CR/LF symbol (crosshatched).

In the three symbol removal techniques, the black pels associated with symbols recognized and coded are "removed" (changed to white), and then the entire document is encoded using the Modified READ code, including the areas where the symbols were. In the other three techniques, the Modified READ code is used only for areas that do not have encoded characters.

3.1 SYMBOL REMOVAL/SCAN LINE

This approach is similar to the Combined Symbol Matching (CSM) Technique presented by Compression Labs, Inc. in APPENDIX B, but with a stored library instead of a transmitted library.

3.1.1 DESCRIPTION

In this approach the document is scanned, from top to bottom, and from left to right, until a group of black pels is encountered that matches a symbol in the stored library. When this occurs, all the black pels within the rectangular symbol space are changed to white, and the symbol code and position are recorded. After the symbols have been removed, the document is rescanned in principle and encoded using the Modified READ code ($K = \infty$, no EOL code). The detected symbol codes are inserted before the READ code of the scan line in which the top of the symbol occurs. The presence of a symbol code, rather than READ code, is indicated by a single bit at the beginning of every scan line. If the bit indicates that there are symbols on the scan line, the 8-bit symbol code follows, and this is followed by an 11-bit horizontal position code word. (2^{11} = 2,048, which is greater than 1,728 pels in a line.) This may be followed by any additional symbols on the scan line (in order of horizontal position), and the symbol data is terminated by a special 8-bit symbol code that indicates that there are no more symbols on the scan line. Then the READ code for that scan line (less encoded symbols) is transmitted. Figure 3-1 illustrates

the composition of a mixed-mode message using this segmentation approach.

Notice that in this technique the recognized symbols will be encoded as they are first encountered by the scanning process, regardless of where they appear relative to other symbols or graphics. The vertical position of the symbols is implied by the scan line on which the symbol code appears.

Figures 3-2 and 3-3 illustrate the areas of each document that are encoded as symbols using the SYMBOL REMOVAL/SCAN LINE technique.

3.1.2 COMPRESSION ESTIMATE

DOCUMENT 1

All of the typewritten symbols, plus the "SAPORS LANE" line will be encoded, which is a total of 830 symbols. (See Figure 3-2). This will take 8 x 830 = 6,640 bits. A single bit at the beginning of each scan line indicates the presence of symbols on the line, which will use 2,376 bits, since there are 2,376 scan lines. It is assumed that the horizontal position of the symbol will require 11 bits. (This could possibly be reduced somewhat by more efficient coding.) This will take 11 x 830 = 9,130 bits. For each of the 202 scan lines on which symbols are detected, a special 8-bit symbol code will be used to indicate the end of symbols and the start of graphics. This will take 8 x 202 = 1,616 bits. Finally the graphics will take 31,828 bits, as discussed in Section 2.2.2.1.

Summarizing:

Symbol codes	6,640
Symbol present	2,376
Horizontal position	9,130
End of symbols	1,616
Graphics	31,828
	51,590 bits

$$\text{Compression} = \frac{2{,}376 \times 1{,}728}{51{,}590} = 79.6$$

DOCUMENT 5

All of the symbols except subscripts and superscripts will be encoded (see Figure 3-3), for a total of 1,599 symbols. This will take 8 x 1,599 = 12,792 bits. The horizontal position will take 11 x 1,599 = 17,589 bits. The end of symbols code will take 8 x 445 = 3,560 bits, and graphics use 37,894 bits (see Section 2.2.2.2).

Summarizing:

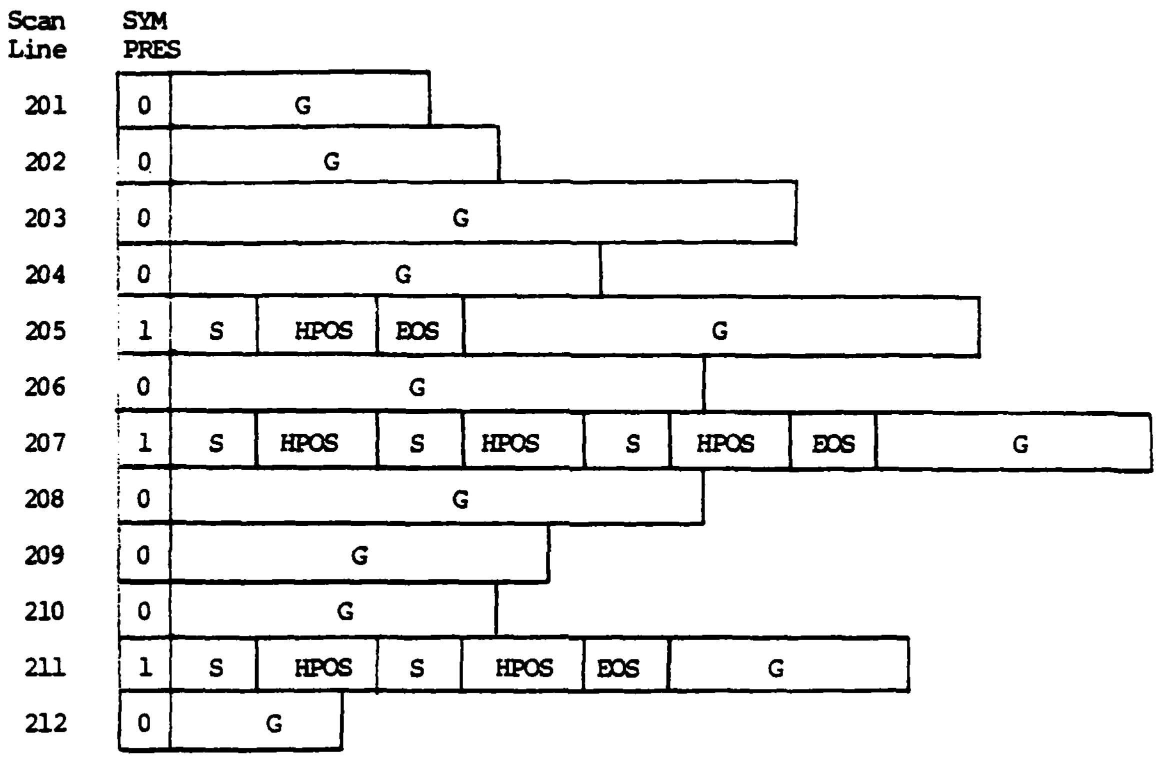
Symbol Codes	12,792
Symbol present	2,376
Horizontal position	17,589
End of symbols	3,560
Graphics	37,894
	74,211 bits

$$\text{Compression} = \frac{2{,}376 \times 1{,}728}{74{,}211} = 55.3$$

Figure 3-1

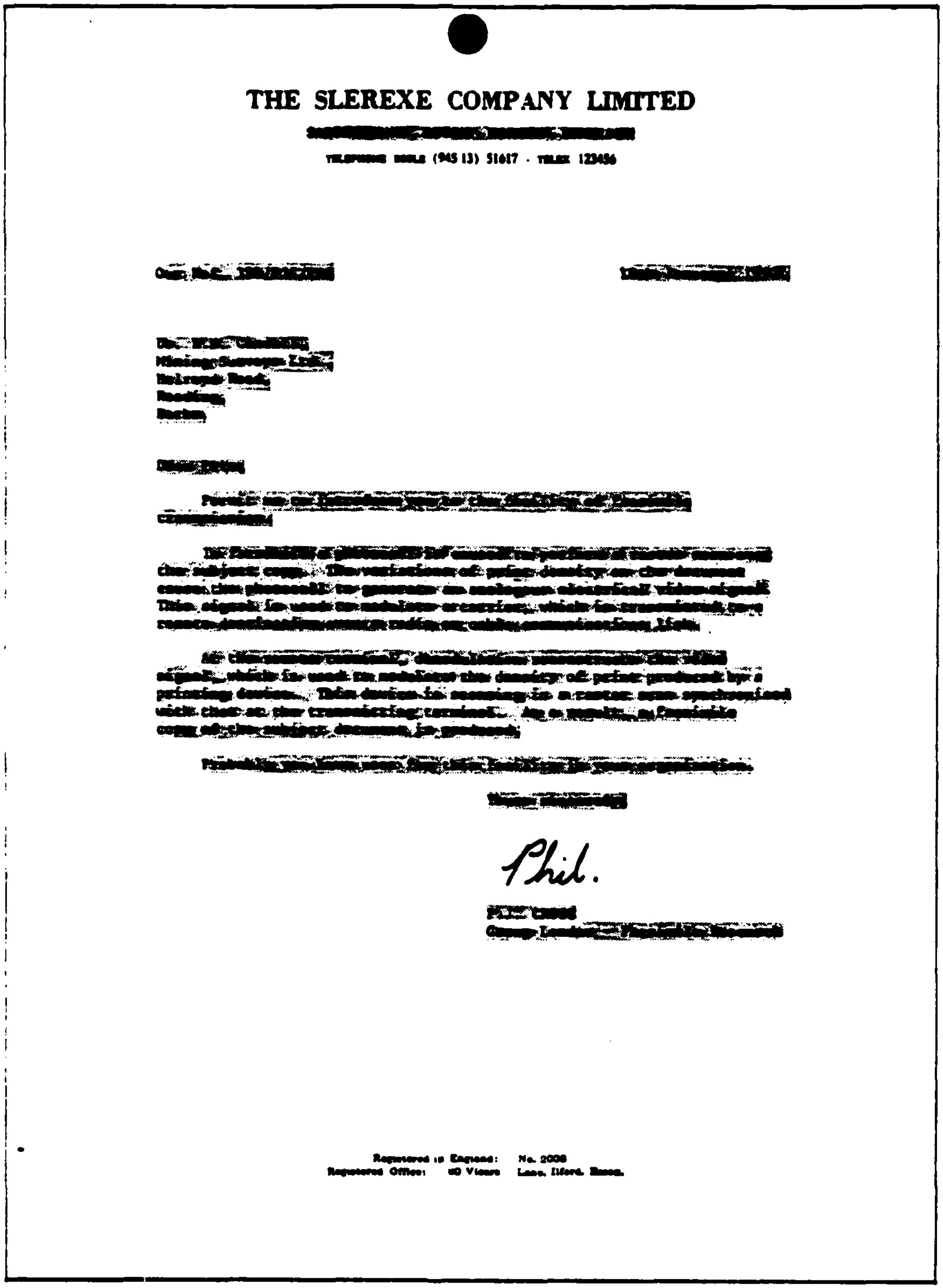
MESSAGE COMPOSITION

SYMBOL REMOVAL/SCAN LINE



LEGEND

SYM PRES	1 indicates at least one symbol on scan line - 1 bit
G	graphics mode using Modified READ code - variable bits
S	symbol code - 8 bits
HPOS	horizontal position of symbol - 11 bits
EOS	end of symbols on scan line - 8 bits



THE SLEREXE COMPANY LIMITED

(945 13) 51617 · 123456

Phil.

Figure 3-2

SYMBOL REMOVAL/SCAN LINE

Cela est d'autant plus valable que [illegible] grand. A cet égard la figure 2 représente la vraie courbe donnant [illegible] en fonction de f [illegible] riques indiquées page précédente.

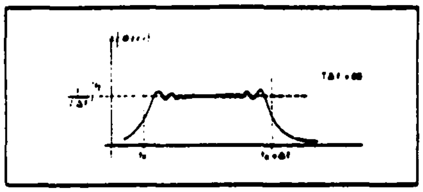


FIG. 2

Donc en ce cas, le filtre [illegible] conformément à la figure 3, [illegible]

— d'un filtre [illegible] $f_0 \leq f \leq f_0 + \Delta f$ [illegible] $f < f_0$ et $f > f_0 + \Delta f$ [illegible] des composantes [illegible]

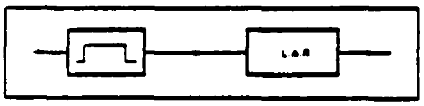


FIG. 3

— [illegible]

$$T_x = T_0 + (f_0 - f)\,\frac{T}{\Delta f}$$

[illegible]

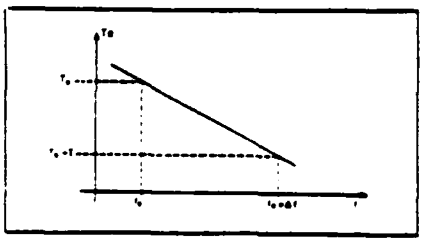


FIG. 4

[illegible]

$$\varphi = -2\pi \int_0^f T_x\, df$$

$$\varphi = -2\pi \left[T_0 + \frac{f_0 T}{\Delta f} \right] f + \pi\,\frac{T}{\Delta f}\,f^2$$

[illegible]

Un signal [illegible] donne à la sortie [illegible] de Fourier est réelle, constante entre f_0 et $f_0 + \Delta f$ [illegible] à-dire un signal de fréquence [illegible] dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de [illegible] compression d'impulsion [illegible] filtre adapté [illegible]

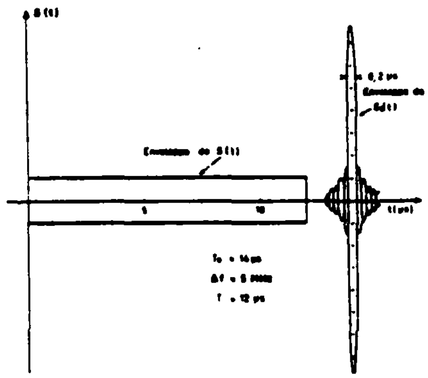


FIG. 5

[illegible]

Figure 3-3

SYMBOL REMOVAL/SCAN LINE DOCUMENT 5

3.2 SYMBOL REMOVAL/FULL DOCUAMENT

This approach is similar to SYMBOL REMOVAL/SCAN LINE, except that the symbols and graphics are not interleaved in the transmitted message.

3.2.1 DESCRIPTION

In this approach the symbols are detected, removed, and their codes and positions recorded, as in SYMBOL REMOVAL/SCAN LINE. But in this technique, all the symbol information is transmitted before any graphics. The vertical position of the symbols is transmitted using an 12-bit code. (2^{12} = 4,096, which is greater than 2,376 scan lines). Then the 8-bit symbol code and 11-bit horizontal position code follows for the first symbol at that vertical position. If more symbols have the same vertical position, their symbol codes and horizontal positions follow. Following all the symbols on the scan line, a special 8-bit symbol is transmitted to terminate the scan line. The symbols on the following scan lines follow in succession. The symbol mode is terminated by a single 12-bit code, possibly a vertical position greater than 2,376. Then the entire document, with the symbols removed, is transmitted by Modified READ code. Figure 3-4 illustrates the composition of a mixed-mode message using the SYMBOL REMOVAL/ FULL DOCUMENT segmentation approach.

Again in this technique the recognized symbols will be encoded regardless of their position. The vertical position

of each symbol is explicitly stated, but the mode change between symbols and graphics must be done only once in the entire document.

The areas encoded as symbols by SYMBOL REMOVAL/FULL DOCUMENT are shown in Figures 3-5 and 3-6.

3.2.2 COMPRESSION ESTIMATE

DOCUMENT 1

Again, 830 symbols will be encoded, taking 8 x 830 = 6,640 bits. For each of the 202 scan lines on which a symbol appears, the vertical position must be given. For a binary code, this will take 12 bits, or 12 x 202 = 2,424 bits. In addition, the last symbol on the scan line will be indicated by a special 8-bit code, taking 8 x 202 = 1,616 bits. For each symbol, the horizontal position is given in an 11-bit binary code, for a total of 11 x 830 = 9,130 bits. The mode change from symbols to graphics occurs only once in the entire image, and takes only 8 bits. Finally, the graphics will take 31,828 bits. Summarizing:

Symbol codes	6,640
Vertical position	2,424
Horizontal position	9,130
Last symbol on scan line	1,616
Symbols to graphics	8
Graphics	31,828
	51,646 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{51,646} = 79.5$$

DOCUMENT 5

All of the symbols except subscripts and superscripts will be encoded, for a total of 1,599 or 8 x 1,599 = 12,792 bits. For each of the 445 scan lines on which a symbol appears, a 12-bit vertical position will be given, using 12 x 445 = 5,340 bits. The code for the last symbol on the scan line will take 8 x 445 = 3,560 bits. The horizontal position for each symbol will take 11 x 1,599 = 17,589 bits. The graphics again take 37,894 bits. Summarizing:

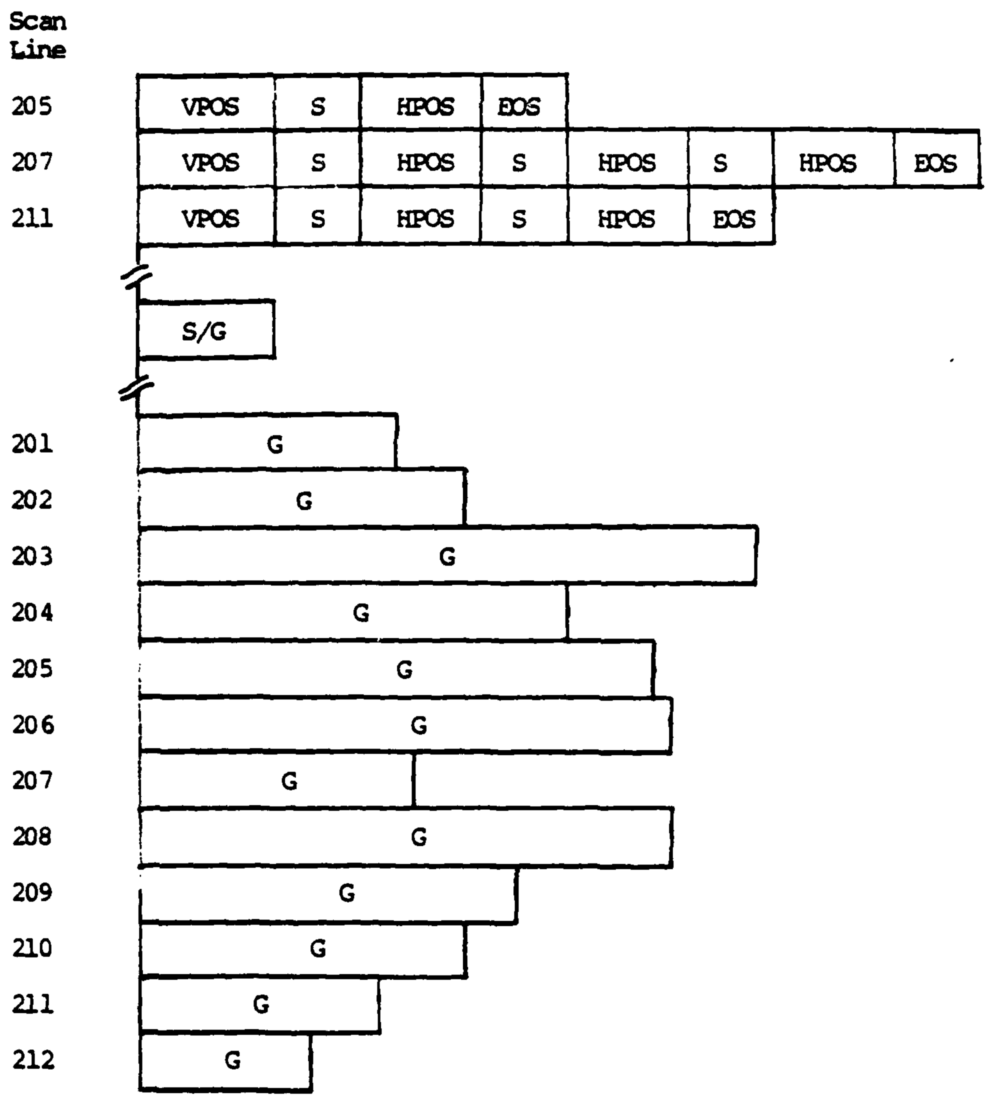
Symbol codes	12,792
Vertical position	5,340
Horizontal position	17,589
Last symbol on scan line	3,560
Symbols to graphics	8
Graphics	37,894
	77,183 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{77,183} = 53.2$$

Figure 3-4

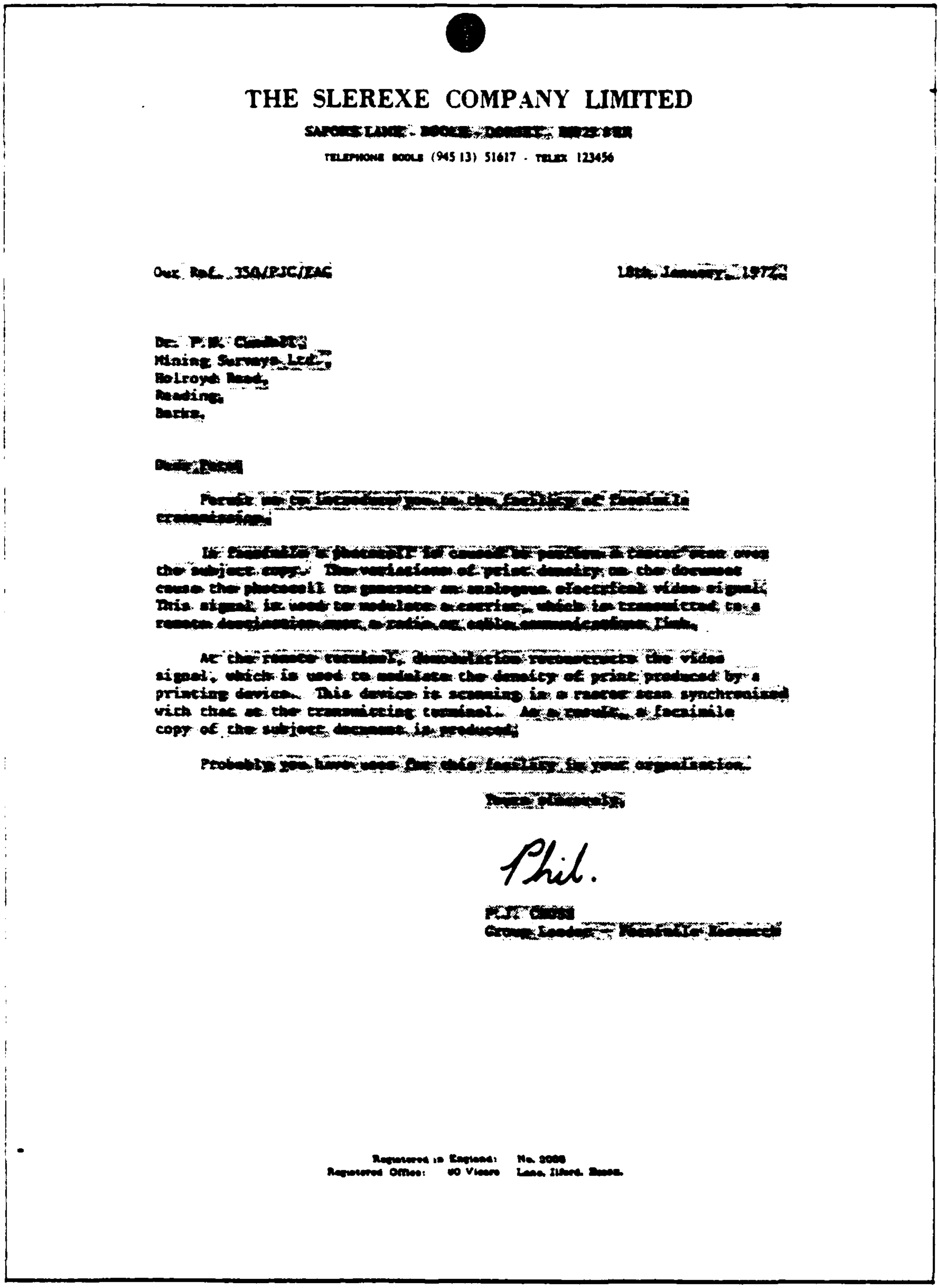
MESSAGE COMPOSITION

SYMBOL REMOVAL/FULL DOCUMENT



LEGEND

G	graphics mode using Modified READ code - variable bits
S	symbol code - 8 bits
VPOS	vertical position of symbols - 12 bits
HPOS	horizontal position of symbol - 11 bits
EOS	end of symbols on scan line - 8 bits
S/G	indicates change from symbols to graphics - 12 bits



THE SLEREXE COMPANY LIMITED

SAPORS LANE - BOOLE - DORSET - BH25 8ER

TELEPHONE BOOLE (945 13) 51617 - TELEX 123456

Our Ref. 350/PJC/EAC 18th January, 1972.

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England: No. 2038
Registered Office: 60 Vicara Lane, Ilford. Essex.

Figure 3-5 SYMBOL REMOVAL/FULL DOCUMENT Document 1

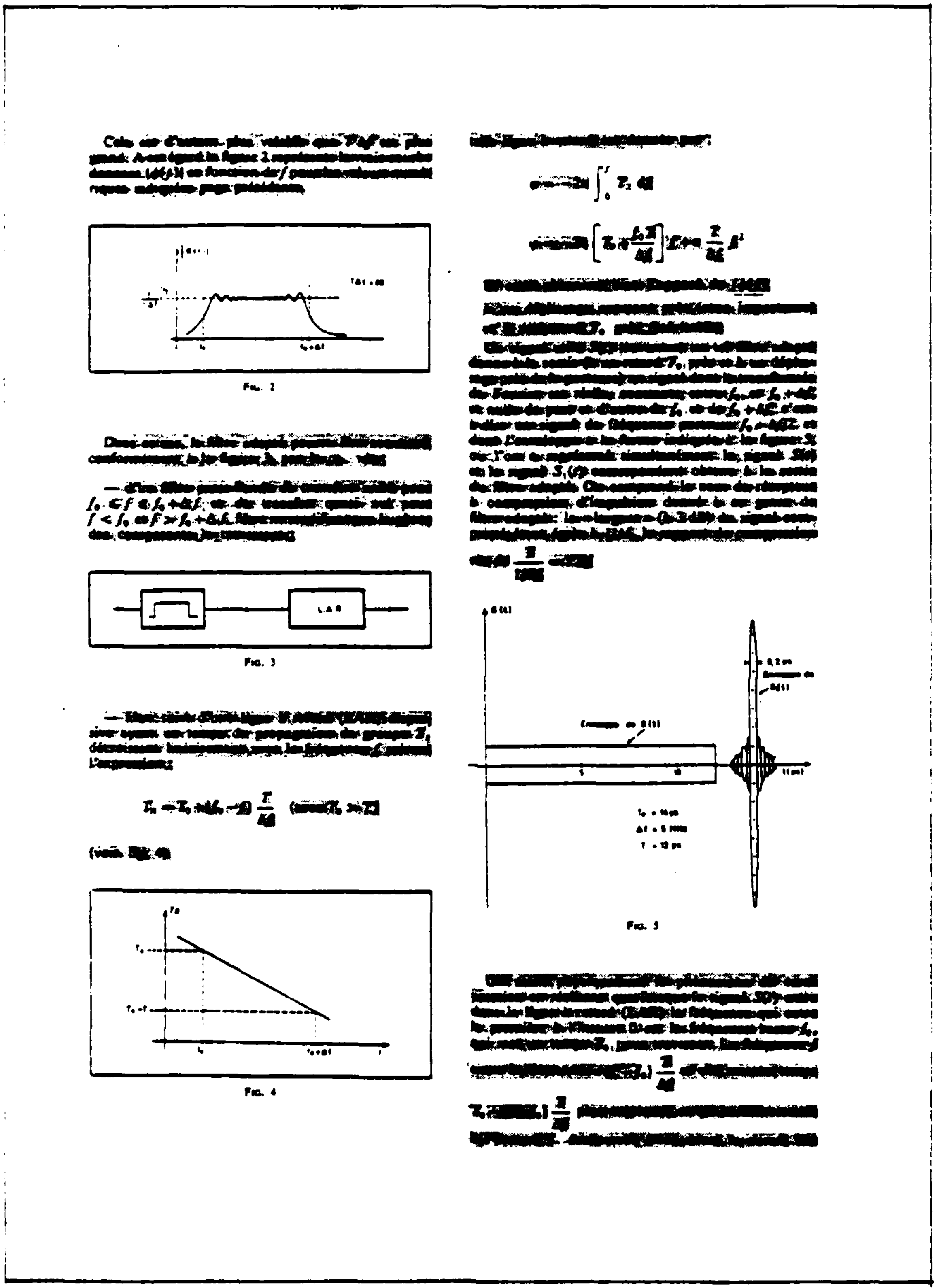


Figure 3- 6

SYMBOL REMOVAL/FULL DOCUMENT   Document 5

3.3 SYMBOL REMOVAL/LINE OF SYMBOLS

3.3.1 DESCRIPTION

In this technique the symbols are detected, removed, and their codes and positions recorded as in the other symbol removal approaches. The symbols are then organized into lines of symbols, based on the symbol position, height, hang down position, etc. Account is taken of small amounts of line skew, and a single vertical position is assigned to the entire line of symbols. When this process is complete, each printed line on the document should be contained within a line of symbols. Spaces between symbols are filled by appropriate blank characters, having several different widths, up to about 2 normal symbol spaces. If the space between symbols is greater than 2 symbol spaces, the line of symbols is broken into segments.

The entire document, less recognized symbols, is transmitted using Modified READ code. When a scan line having the vertical position of a line of recognized symbols is encountered, a special 12-bit code (which could be an EOL code) is inserted. This changes the mode from graphics to symbols. This is followed by an 11-bit code giving the horizontal position of the first symbol. Then the symbol codes for each symbol in the segment are sent, followed by a special 8-bit end-of-segment symbol code. This is followed by an 11-bit distance to the next segment of symbols. The last segment of symbols on the line is followed by a special 8-bit end-of-line symbol code instead of the end-of-segment code. This changes the mode back to graphics, and the Modified READ code is continued, until another scan line with a line of symbols

is encountered. Figure 3-7 illustrates the composition of a mixed-mode message using the SYMBOL REMOVAL/LINE OF SYMBOLS segmentation technique.

As with the other symbol removal techniques, a recognized symbol will be encoded wherever it is located, since lines of symbols may overlap vertically, and each line of symbols may have as few as one symbol. There may be some problem in accurately positioning symbols, since spaces between symbols of 1 or 2 pels will probably not be encoded, and the horizontal position of a symbol could be in error at the end of a long line of symbols.

The symbols encoded by the SYMBOL REMOVAL/LINE OF SYMBOLS technique are shaded in Figures 3-8 and 3-9, and the lines of symbols are enclosed by lines.

3.3.2 COMPRESSION ESTIMATE

DOCUMENT 1

There are 966 symbols to be encoded (includes spaces between words), taking 8 x 966 = 7,728 bits. There are 24 lines of symbols with 25 segments. Each line has a 12-bit code to indicate that there are symbol segments on the line, or 12 x 24 = 288 bits. The horizontal position of the first symbol is given by an 11-bit binary code, or 11 x 24 = 264 bits. The distance between segments is also given by a 11-bit binary code. There is only one of these for 11 bits total. There are two special 8-bit codes for end of segment and end of line. There is one end of segment code for 8 bits, and 24 x 8 = 192 bits for end of line codes. Summarizing:

Symbol codes	7,728
Symbols on line	288
Horizontal position	264
Distance between segments	11
End of segment	8
End of line	192
Graphics	31,828
	40,319 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{40,319} = 101.8$$

DOCUMENT 5

There are 1,887 symbols to be encoded (including spaces between words), taking 8 x 1,887 = 15,096 bits. It is assumed that subscripts break a line into segments, since they are not recognized and cannot be removed. Single character lines can be encoded. There are 46 lines with 90 segments. The lines with symbols are indicated by 12 x 46 = 552 bits. The horizontal position of the first symbol is indicated by 11 x 46 = 506 bits. The distance between segments is given by 11 x (90-46) = 484 bits. There are 8 x (90-46) = 352 bits for end of segment and 8 x 46 = 368 bits for end of line. Graphics again takes 37,894 bits as given in Section 2.2.2.2. Summarizing:

Symbol codes	15,096
Symbols on line	552
Horizontal position	506
Distance between segments	484
End of segment	352

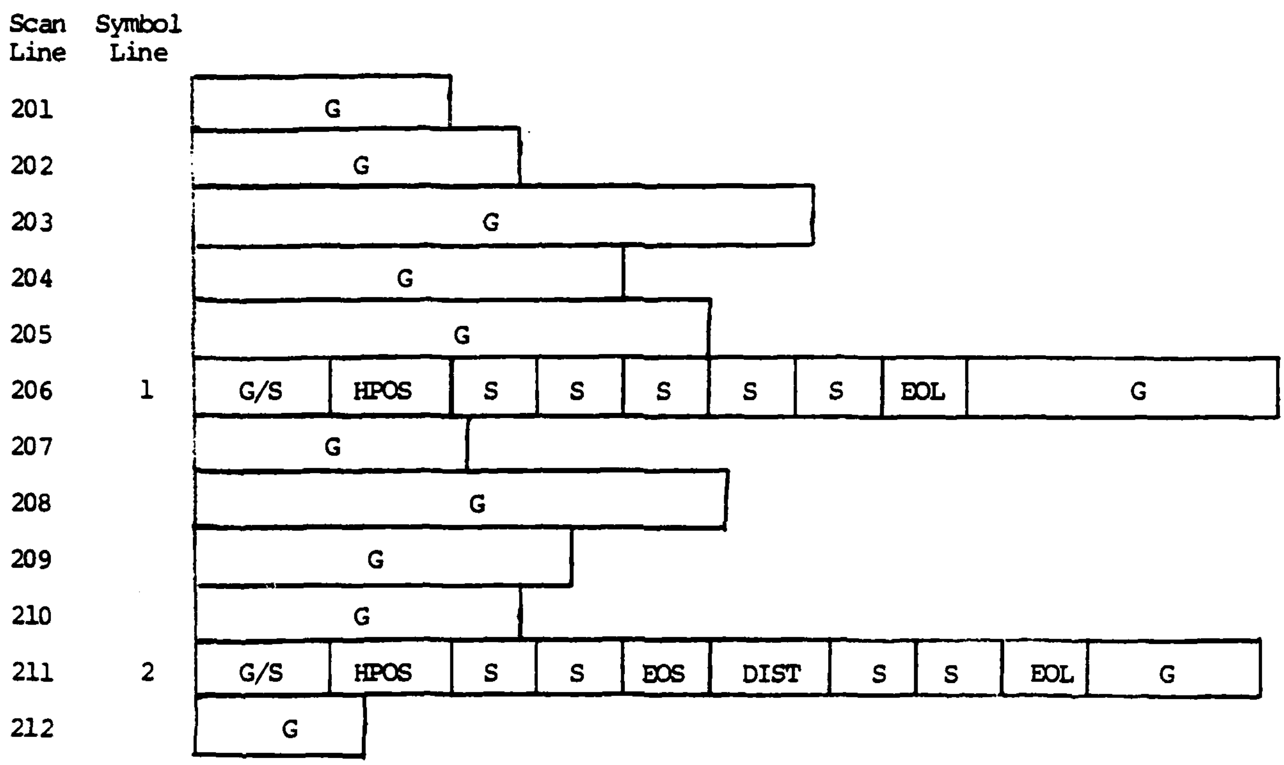
End of line	368
Graphics	<u>37,894</u>
	55,252 bits

$$\text{Compression} = \frac{2{,}376 \times 1{,}728}{55{,}252} = 74.3$$

Figure 3-7

MESSAGE COMPOSITION

SYMBOL REMOVAL/LINE OF SYMBOLS



<u>LEGEND</u>

G	graphics mode using Modified READ code - variable bits
S	symbol code - 8 bits
G/S	indicates change from graphics to symbols - 12 bits
HPOS	horizontal position of first symbol in line - 11 bits
EOL	end of symbols on line - 8 bits
EOS	end of symbols on segment - 8 bits
DIST	distance between segments - 11 bits

THE SLEREXE COMPANY LIMITED

SAPORS LANE . POOLE . DORSET . BH 25 8ER

TELEPHONE POOLE (945 13) 51617 · TELEX 123456

Our Ref. 350/PJC/EAC   18th January, 1972

Dr. P.N. Cundall,
Mining Surveys Ltd.,
Holroyd Road,
Reading,
Berks.

Dear Pete,

Permit me to introduce you to the facility of facsimile transmission.

In facsimile a photocell is caused to perform a raster scan over the subject copy. The variations of print density on the document cause the photocell to generate an analogous electrical video signal. This signal is used to modulate a carrier, which is transmitted to a remote destination over a radio or cable communications link.

At the remote terminal, demodulation reconstructs the video signal, which is used to modulate the density of print produced by a printing device. This device is scanning in a raster scan synchronised with that at the transmitting terminal. As a result, a facsimile copy of the subject document is produced.

Probably you have uses for this facility in your organisation.

Yours sincerely,

Phil.

P.J. CROSS
Group Leader - Facsimile Research

Registered in England No. 2038
Registered Office ... Vicars Lane, Ilford. Essex.

Figure 3-8

SYMBOL REMOVAL/LINE OF SYMBOLS Document 1

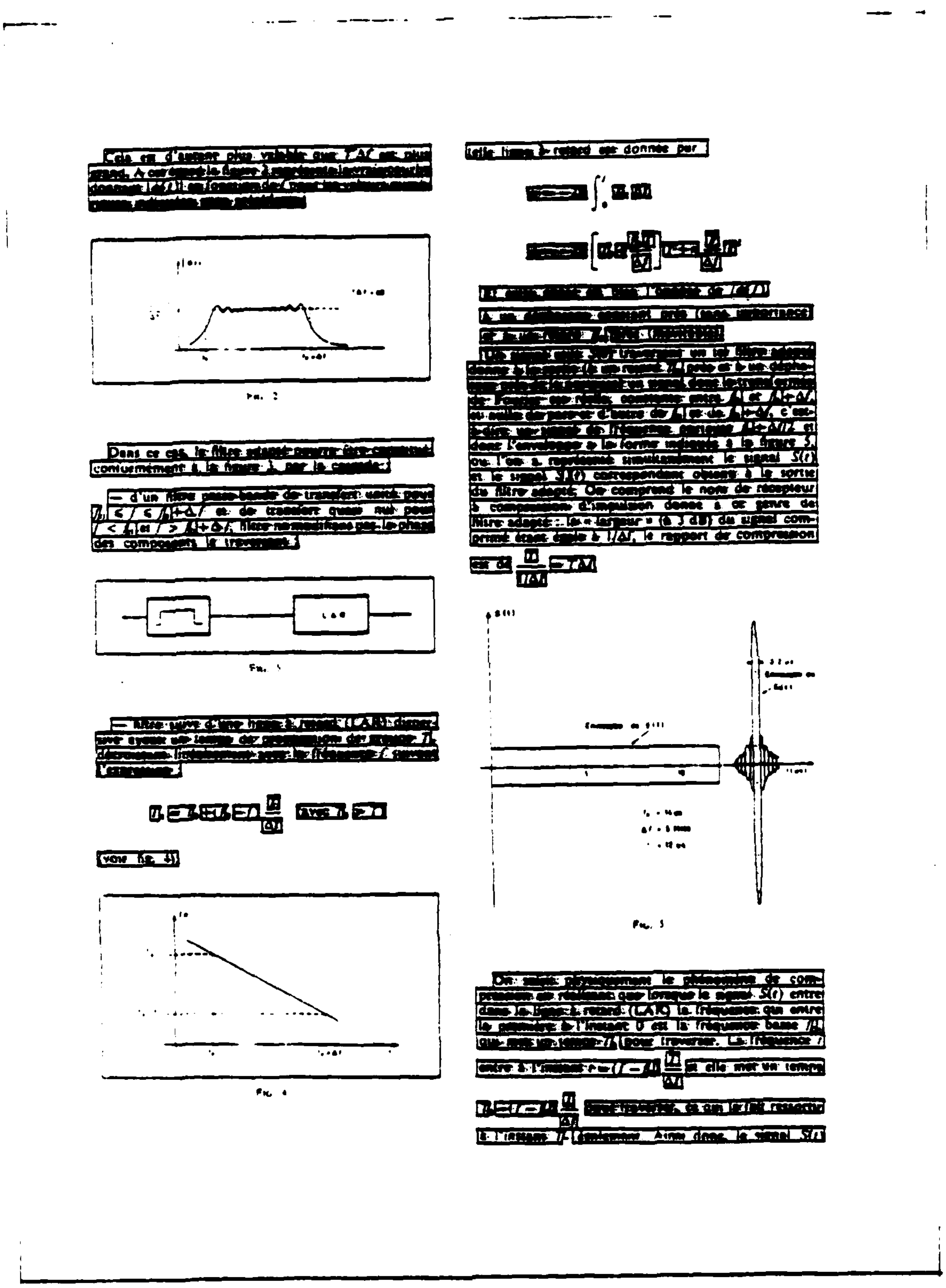


Figure 3- 9

SYMBOL REMOVAL/LINE OF SYMBOLS Document 5

3.4 EXTENDED TELETEX

This approach was suggested by MBLE, (Appendix C), using the basic TELETEX method to transmit the entire document, except where there are graphic areas that must be transmitted by facsimile.

3.4.1 DESCRIPTION

In this approach the entire document is divided into character spaces, except for areas that are defined as graphics, as discussed below. All character spaces, including blanks, are transmitted using 8-bit symbol codes.

The graphics are transmitted by Modified READ code as they occur within a line of symbols. First, a special 8-bit symbol code is used to designate the transition from symbol codes to graphics. This is followed by an 11-bit code giving the width of the graphics area. (The height of the graphics area is defined by the height of the symbol font.) Then the READ code for the graphics is sent. The length of the READ code is defined by the width and height of the graphics area, so the transition back to symbol codes does not require a code.

As an option, instead of transmitting a series of blank symbol codes at the right end of the line, a special 8-bit code could be used to designate the last symbol on the line. This would also have to be to the right of any graphics on the line. This symbol would direct the receiver to start on the next line of symbols, and would replace the CR and LF codes of TELETEX. For reasons of commonality it may be preferred to keep the two

standard TELETEX symbols for this purpose. Figure 3-10 illustrates the composition of a message using the EXTENDED TELETEX segmentation technique.

This technique is considered primarily as a method to incorporate graphics (such as logos and signatures), into computer-generated text. Therefore graphics areas are defined, probably by the user, as rectangular areas which may contain a mixture of graphics and symbols. As a result, it may be difficult for a scanner to block out a difficult document such as Document 5.

Since all lines of symbols must have the proper spacing, symbols that are not aligned with the majority of the symbols must be treated as graphics.

If lines of symbols exceed their normal vertical spacing, as in Document 5, a graphics filler can be inserted with any number of scan lines to keep the symbol grid aligned with the symbols on the document. This could be initiated with a special symbol code.

The symbols encoded by the EXTENDED TELETEX technique are shaded in Figures 3-11 and 3-12, with the cross-hatched area indicating space symbols that would not be transmitted if the CR/LF symbol is used.

3.4.2 COMPRESSION ESTIMATE

DOCUMENT 1

Character space is $\frac{1}{6}$" high by $\frac{1}{12}$" wide, or 32.6 pels high by 16.3 pels wide. The number of symbol lines is

$\frac{2,376}{32.6}$ = 73 and number of symbols per line is $\frac{1,728}{16.3}$ = 106. Total symbol spaces on the entire document is 73 x 106 = 7,738. From this must be subtracted the symbol spaces in graphics areas shown in Figure 3-11, 251, or 7,487 character spaces. With 8 bits per symbol, 8 x 7,487 = 59,896 bits are required. Only 12 symbol lines have graphics. Each of these lines will need an 8-bit symbol-to-graphics code, 8 x 12 = 96 bits, and an 11-bit graphics width code (11 x 12 = 132). Bits to transmit graphics area will be the same as before, 31,828. Summarizing:

Symbol codes	59,896
Symbol to graphics	96
Graphics width	132
Graphics	31,828
	91,952 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{91,952} = 44.7$$

With a CR/LF code, many space characters can be saved. In the lines with symbols, 962 symbols are saved, in 37 blank lines 3,922 symbols, and in graphics lines 533 symbols, for a total of 5,417 symbols, or 8 x 5,417 = 43,336 bits. Subtracting this from the symbol codes leaves a net of 59,896 - 43,336 = 16,560 bits. To this must be added an 8-bit CR/LF code for each of 73 lines, or 8 x 73 = 584 bits. Summarizing:

Symbol codes	16,560
Symbols to graphics	96
Graphics width	132
CR/LF	584
Graphics	31,828
	49,200 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{49,200} = 83.4 \text{ with CR/LF}$$

DOCUMENT 5

The character space is about 0.148" high by 0.059" wide, or 29 pels high by 11.5 pels wide. The number of symbol lines is $\frac{2,376}{29} = 82$ and the number of symbols per line is $\frac{1,728}{11.5} = 150$. The total number of symbol spaces on the document is 82 x 150 = 12,300, less symbol spaces in the graphics area, 3,207, or 9,093. Using 8 bits per symbol, the symbol codes take 8 x 9,093 = 72,744 bits. In 96 cases a shift from symbols to graphics is required. The symbols to graphics code will take 8 x 96 = 768 bits, and the graphics width code will take 11 x 96 = 1,056 bits. To transmit the graphics area there will be 37,894 bits, plus 138.992 bits for each of the 30 symbols not encoded because they are in a graphics area, or 4,170 bits, for a total of 42,064 bits. Summarizing:

Symbol codes	72,744
Symbol to graphics	768
Graphics width	1,056
Graphics	42,064
	116,632 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{116,632} = 35.2$$

Using the CR/LF, about 2,828 blank characters on the right hand margin could be eliminated. This would reduce the symbol codes by 8 x 2,828 = 22,624 bits to 50,120 bits. There would have to be a CR/LF code for each line, or 8 x 82 = 656 bits. Summarizing:

Symbol codes	50,120
Symbol to graphics	768
Graphics width	1,056
CR/LF	656
Graphics	42,064
	94,664 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{94,664} = 43.4 \text{ with CR/LF}$$

Figure 3-10

MESSAGE COMPOSITION

EXTENDED TELETEX

Symbol Line											
5	S	S	S	CR/LF							
6	S	S	S	S	S	S	S	S	S	CR/LF	
7	S	S	S	S/G	Width	G	CR/LF				
8	S	S	S	S	S	S	S	CR/LF			
9	S/G	Width	G	S	S	S	CR/LF				
10	S	S	S	S	S	S	S	S	CR/LF		
11	S	S	S	S	S	S	S	CR/LF			
12	S/G	Width	G	S	S/G	Width	G	S	CR/LF		
13	S	S	S	S	CR/LF						

LEGEND

G	graphics mode using Modified READ code - variable bits
S	symbol code - 8 bits
CR/LF	carriage return/line feed - 8 bits
S/G	indicates change from symbols to graphics - 8 bits
Width	width of graphics - 11 bits

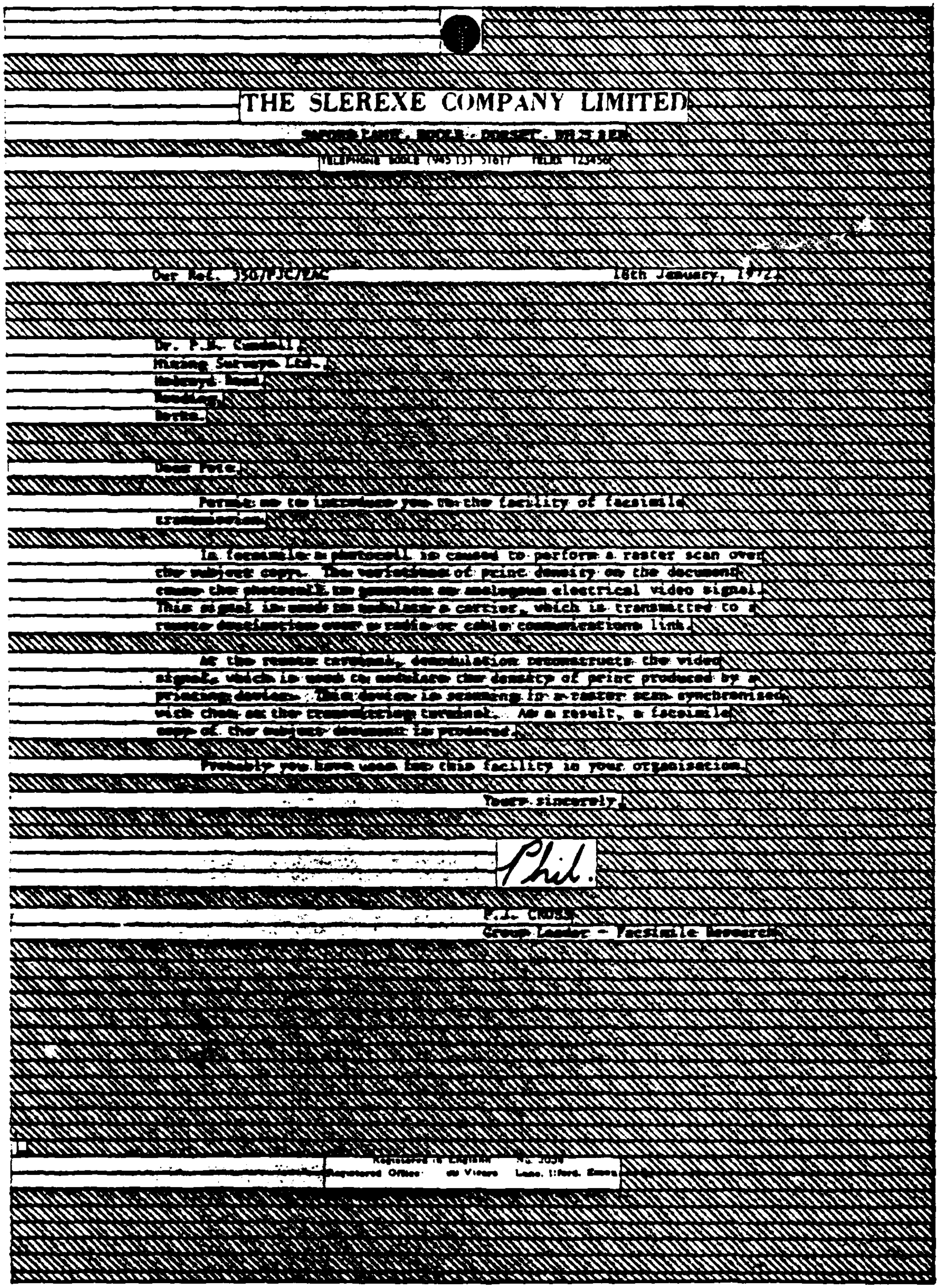


Figure 3-11

EXTENDED TELETEX Document 1

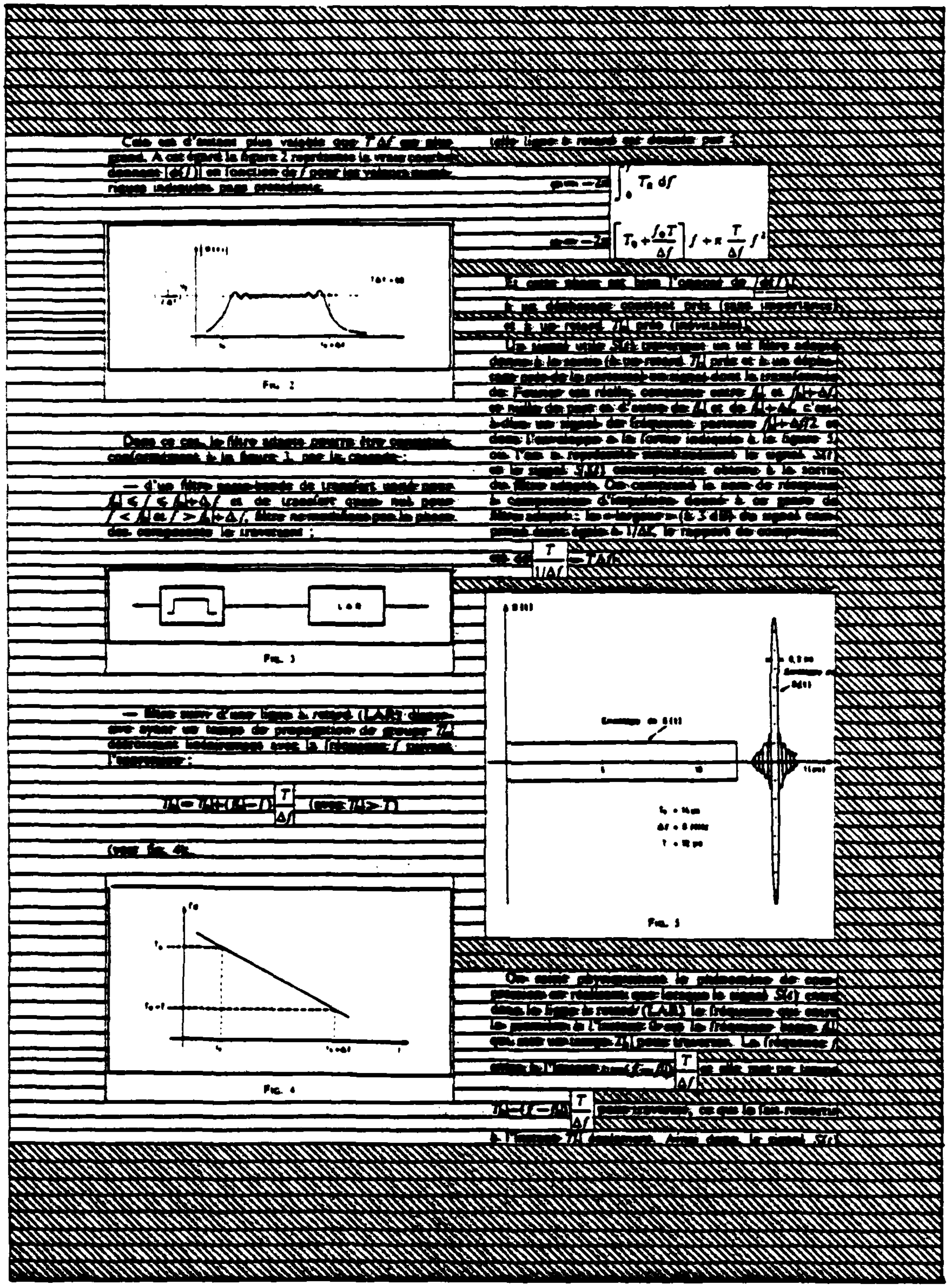


Figure 3-12

EXTENDED TELETEX Document 5

3.5 PARTIAL LINE OF SYMBOLS

3.5.1 DESCRIPTION

In this approach, the document is searched for recognizable symbols, which are then organized into lines, as in SYMBOL REMOVAL/LINE OF SYMBOLS. The symbol codes and horizontal positions are stored, together with the vertical position of the line. Then the document is transmitted, basically using Modified READ code. When a line of symbols is encountered, a 12-bit (EOL) code is used to designate the line as containing symbols. The entire line of symbols including blanks, is transmitted using symbol codes, unless there is any graphic material contained within the boundaries of the line. The end of the line is indicated when the total of the widths of the symbols equals 1,728.

If graphics are contained somewhere within the line, a special 8-bit code is used to indicate a change to READ code. This is followed by an 11-bit code giving the width of the graphics material. (The width could be to the end of line if there are no more symbols on the line; changes between graphic and symbol mode are made only when necessary.) This is followed by Modified READ code defining the graphics area within the boundaries of the line. The length of READ code is defined by the height of the line (constant for the font) and the width of the graphic area. This is then followed by more symbols, unless the end of the line has been reached.

As an option, trailing blanks in the line can be re-

placed by a single 8-bit code indicating the last symbol on the line. This code tells the receiver that the line is complete, and the next line, or scan line, will be sent next. Figure 3-13 illustrates the composition of a message using the PARTIAL LINE OF SYMBOLS segmentation technique.

Lines of symbols cannot overlap vertically in this technique, so symbols out of vertical alignment must be treated as graphics.

The symbols encoded by the PARTIAL LINE OF SYMBOLS technique are shaded in Figures 3-14 and 3-15 with the cross-hatched area indicating blank symbols that would not be transmitted if the CR/LF symbol is used.

3.5.2 COMPRESSION ESTIMATE

DOCUMENT 1

There are 24 lines of symbols and each line has 106 symbol spaces, so the number of symbols is 24 x 106 = 2,544 which will take 2,544 x 8 = 20,352 bits. Each symbol line has a 12-bit indicator, which is 24 x 12 = 288 bits. No lines of symbols have interleaved graphics, so there are no symbol to graphics bits. Summarizing:

Symbol codes	20,352
Symbol indicator	288
Last symbol	0
Graphic distance	0
Graphics	31,828
	52,468 bits

$$\text{Compression} = \frac{2{,}376 \times 1{,}728}{52{,}468} = 78.3$$

If a CR/LF symbol is used, 928 blank symbols are saved, so the total number of symbols is 2,496 - 928 = 1,568, which take 8 x 1,568 = 12,544 bits. To this must be added 24 CR/LF symbols, for 8 x 24 = 192 bits. Summarizing:

Symbol codes	12,544
Symbol indicator	288
Last symbol	0
Graphic distance	0
CR/LF	192
Graphics	31,828
	44,852 bits

$$\text{Compression} = \frac{2{,}376 \times 1{,}728}{44{,}852} = 91.5 \text{ with CR/LF}$$

DOCUMENT 5

There are 36 lines that can be formed into lines of symbols, and they contain 3,157 symbols, including spaces. This takes 8 x 3,157 = 25,256 bits. There are 36 symbol indicator codes @ 12 bits, or 36 x 12 = 432 bits. There are 14 lines that begin with graphics, and each requires a last symbol code. In addition, there are 39 symbol segments that terminate with graphics, rather than the end of the line. Each of these requires a last symbol code, for a total of 53 codes, taking 8 x 53 = 424 bits. Within the lines of symbols there are 53 segments of graphics, each requiring an 11-bit width code, for a total of 11 x 53 = 583 bits. In this case there are 21 symbols not encoded because they do not lie on the lines of symbols. They will take 138.992 x 21 = 2,919

bits. Added to the 37,894 bits for graphics gives a total of 40,813 bits for graphics.

Symbol codes	25,256
Symbol indicator	432
Last symbol	424
Graphic distance	583
Graphics	40,813
	67,508 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{67,508} = 60.8$$

Using a CR/LF code, 469 blank symbols at the end of 27 lines can be saved. This will reduce the symbol codes by 8 x 469 = 3,752 bits, to 21,504 bits. To this must be added 27 CR/LF codes, using 8 x 27 = 216 bits. Summarizing:

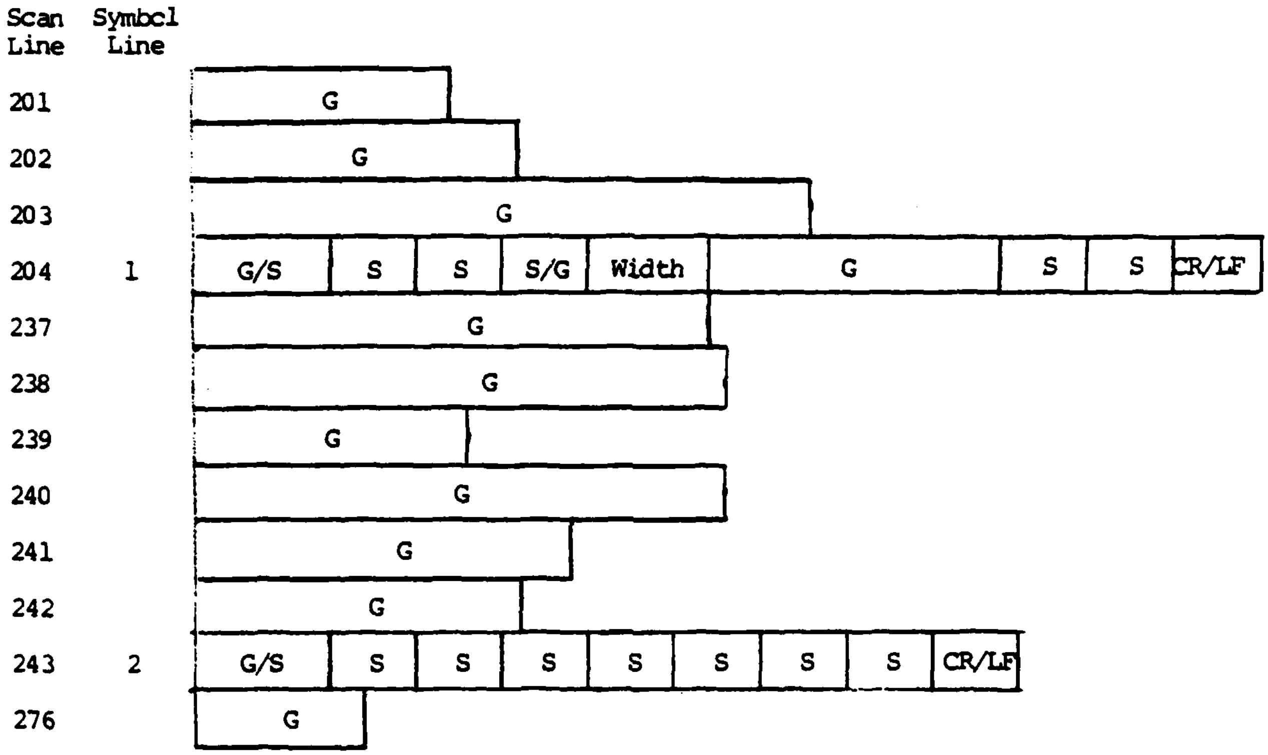
Symbol codes	21,504
Symbol indicator	432
Last symbol	424
Graphic distance	583
CR/LF	216
Graphics	40,813
	63,972 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{63,972} = 64.2 \text{ with CR/LF}$$

Figure 3-13

MESSAGE COMPOSITION

PARTIAL LINE OF SYMBOLS



<u>LEGEND</u>

G	graphics mode using Modified READ code - variable bits
S	symbol code - 8 bits
G/S	indicates change from graphics to symbols - 12 bits
S/G	indicates change from symbols to graphics - 8 bits
CR/LF	carriage return/Line feed - 8 bits
Width	width of graphics - 11 bits

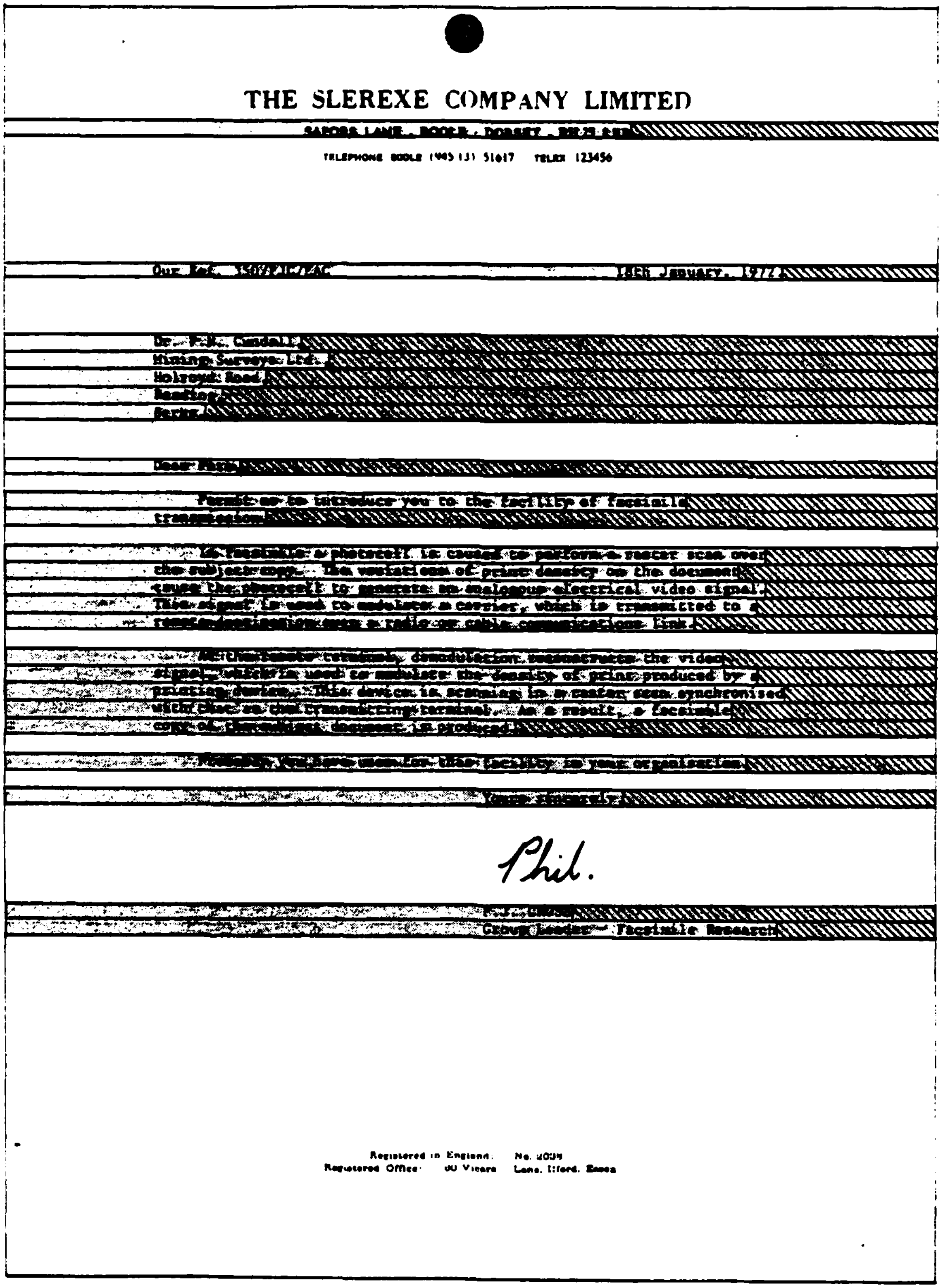


Figure 3-14

PARTIAL LINE OF SYMBOLS Document 1

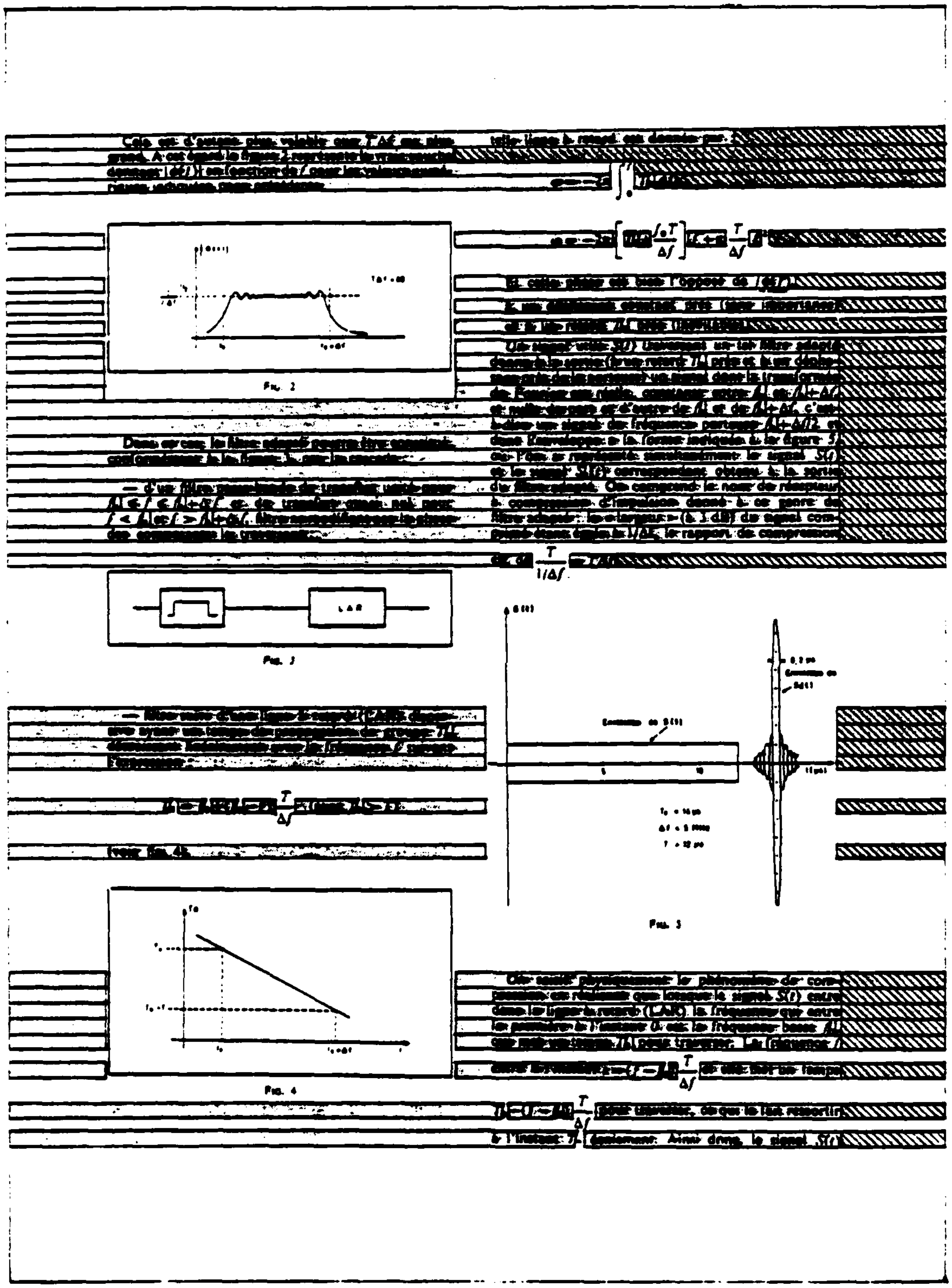


Figure 3-15 PARTIAL LINE OF SYMBOLS Document 5

3.6 FULL LINE OF SYMBOLS

3.6.1 DESCRIPTION

In this approach, the document is searched for recognizable symbols, which are then organized into lines, as in SYMBOL REMOVAL/LINE OF SYMBOLS. The symbol codes and horizontal positions are stored, together with the vertical position of the line. Then the document is transmitted, basically using the Modified READ code. When a line of symbols only is encountered, a 12-bit (EOL) code is used to designate the line as a symbol line. The entire line, including blanks, is transmitted using symbol codes. If any graphics are contained within the boundaries of the line, the entire line is transmitted using the Modified READ code.

Successive lines of symbols are transmitted without a requirement for a mode change, until graphics are detected. At that point, a special 8-bit code is used to signify a return to the graphics mode.

As an option, trailing blanks in the line can be replaced by a single 8-bit code indicating the last symbol on the line. This code tells the receiver that the line is complete, and the next line, or scan line, will be sent next. Figure 3-16 illustrates the composition of a message using the FULL LINE OF SYMBOLS segmentation technique.

Lines of symbols cannot overlap vertically in this technique, so symbols out of vertical alignment must be treated as graphics.

The symbols encoded are shaded in Figures 3-17 and 3-18 with the cross-hatched area indicating blank symbols that would not be transmitted if the CR/LF symbol is used.

3.6.2 COMPRESSION ESTIMATE

DOCUMENT 1

There are 24 lines of symbols, each with 106 symbol spaces, so the number of symbols is 24 x 106 = 2,544, which will take 8 x 2,544 = 20,352 bits. There are 10 groupings of full lines (blank lines are sent by graphics), each requiring a 12-bit graphics-to-symbols indicator, or 12 x 10 = 120 bits. At the end of each grouping of lines there is a 8-bit symbols-to-graphics code, of 8 x 10 = 80 bits. The graphics are 31,828 bits. Summarizing:

Symbol codes	20,352
Graphics to symbols	120
Symbols to graphics	80
Graphics	31,828
	52,380 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{52,380} = 78.4$$

Using the CR/LF character, 928 blank symbols can be saved, reducing the symbol bits by 8 x 928 = 7,424 bits to 12,928 bits, with the addition of 8 x 24 = 192 bits for CR/LF codes. Summarizing:

Symbol codes	12,928
Graphics to symbols	120
Symbols to graphics	80
CR/LF	192
Graphics	31,828
	45,148 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{45,148} = 90.9 \text{ with CR/LF}$$

DOCUMENT 5

Only 6 lines, having 431 symbols, do not have graphics (including subscripts) associated with them, and can be coded as a full line of symbols. Even these lines could not be coded as symbols with the document in its unmodified state, because of the vertical line down the center of the page. There are 6 x 150 = 900 symbol spaces to be encoded, which will take 8 x 900 = 7,200 bits. The full lines appear in 4 groupings. Each change from graphics to symbcls requires a 12-bit code, or 4 x 12 = 48 bits. Each change back to graphics requires an 8-bit code, or 8 x 4 = 32 bits. There are 1,599-431-21= 1,147 additional symbols that are not coded as symbols, and these will take 138.992 bits each, or 1,147 x 138,992 = 159,424 bits as graphics. In addition, the normal graphics takes 40,813 bits, (same as PARTIAL LINE OF SYMBOLS) for a total of 200,237 bits. Summarizing:

Symbol codes	7,200
Graphics to symbols	48
Symbols to graphics	32
Graphics	200,237
	207,517 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{207,517} = 19.8$$

Using the CR/LF code, 144 blank spaces will be saved on the 6 full lines. This reduces the number of symbol spaces to 900 - 144 = 756, which takes 8 x 756 = 6,048 bits. In addition, 6 CR/LF codes are needed, with 8 x 6 = 48 bits. Summarizing:

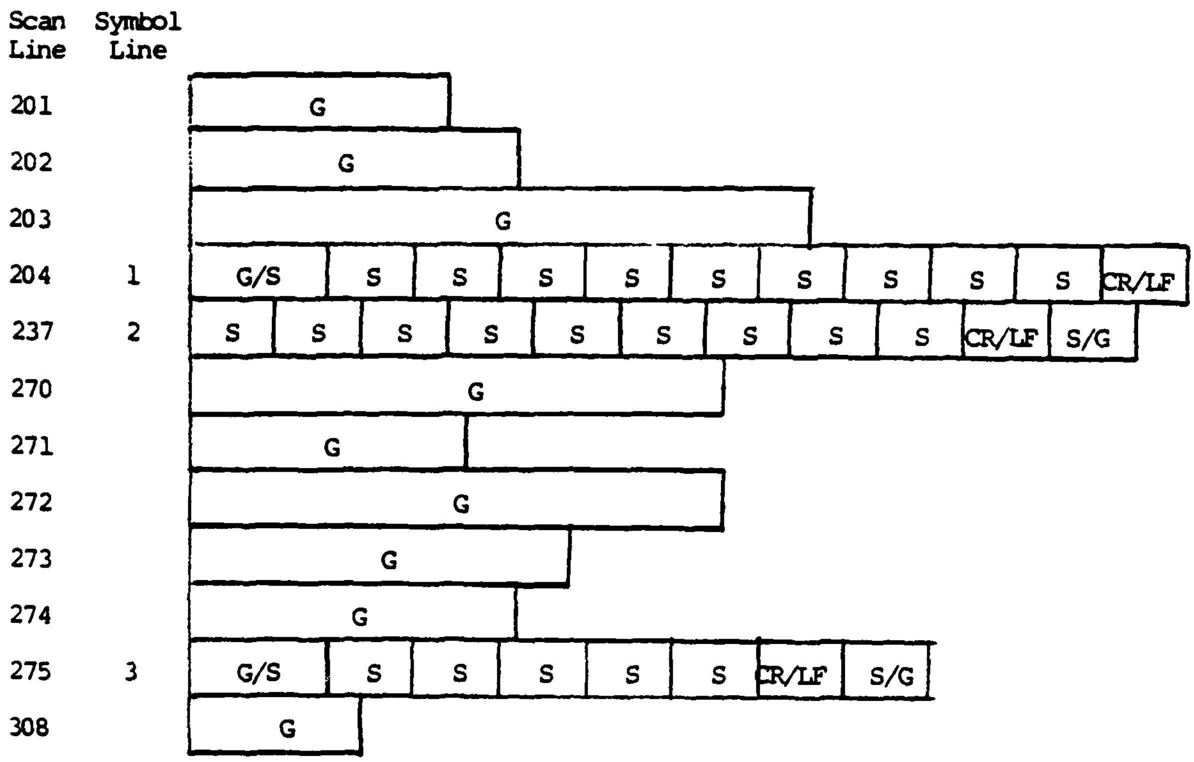
Symbol codes	6,048
Graphics to symbols	48
Symbols to graphics	32
CR/LF	48
Graphics	200,237
	206,413 bits

$$\text{Compression} = \frac{2,376 \times 1,728}{206,413} = 19.9 \text{ with CR/LF}$$

Figure 3-16

MESSAGE COMPOSITION

FULL LINE OF SYMBOLS



LEGEND

G graphics mode using Modified READ code - variable bits
S symbol code - 8 bits
G/S indicates change from graphics to symbols - 12 bits
S/G indicates change from symbols to graphics - 8 bits
CR/LF carriage return/line feed - 8 bits

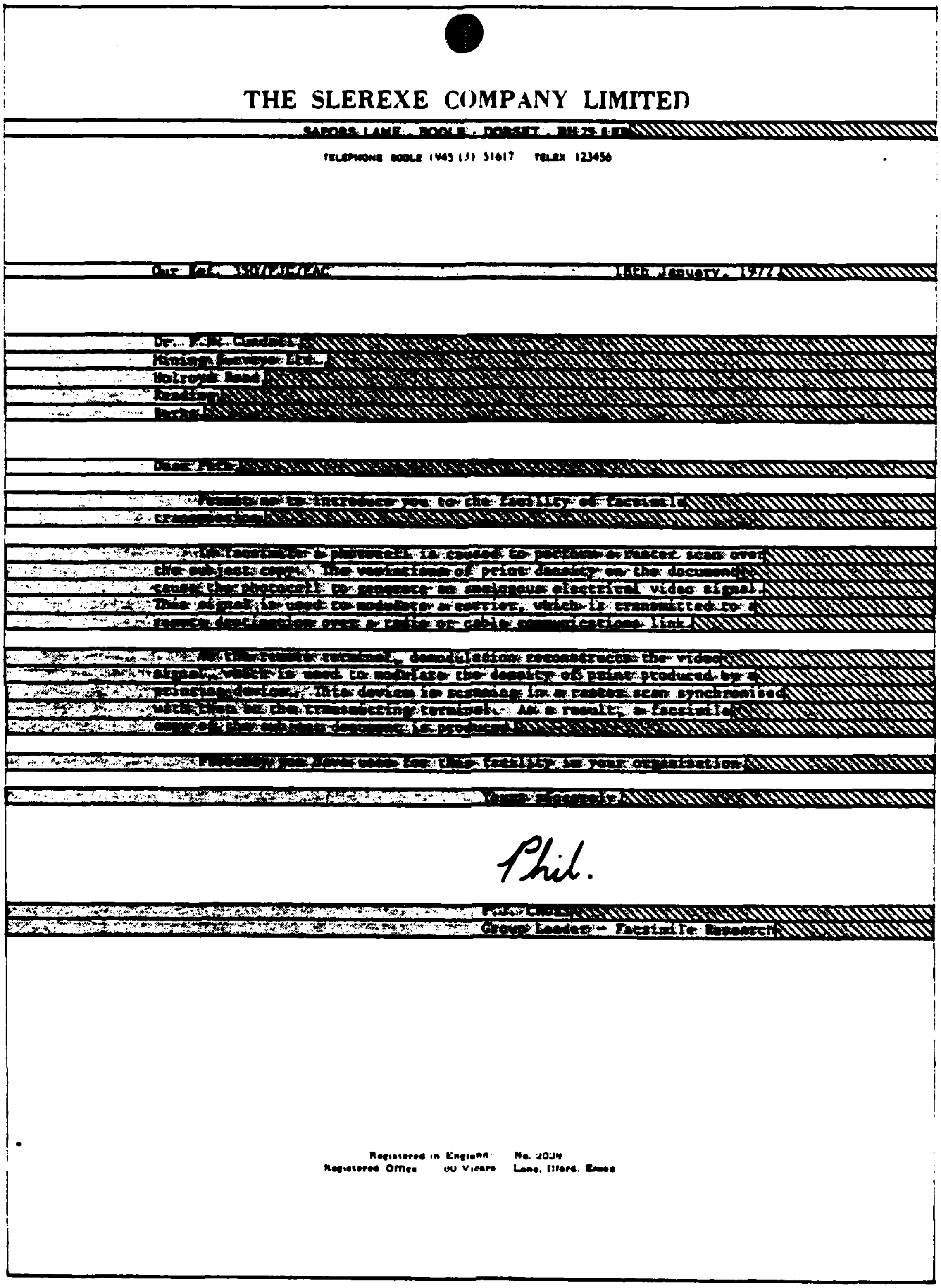


Figure 3-17 FULL LINE OF SYMBOLS Document 1

Cela est d'autant plus valable que $T\Delta f$ est plus grand. A cet égard la figure 2 représente la vraie courbe donnant $|\phi(f)|$ en fonction de f pour les valeurs numériques indiquées page précédente.

FIG. 2

Dans ce cas, le filtre adapté pourra être constitué, conformément à la figure 3, par la cascade :

— d'un filtre passe-bande de transfert unité pour $f_0 \leq f \leq f_0+\Delta f$ et de transfert quasi nul pour $f < f_0$ et $f > f_0+\Delta f$, filtre ne modifiant pas la phase des composantes le traversant ;

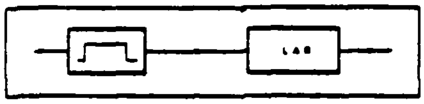


FIG. 3

— filtre suivi d'une ligne à retard (LAR) dispersive ayant un temps de propagation de groupe T_g décroissant linéairement avec la fréquence f suivant l'expression :

$$T_g = T_0+(f_0-f)\frac{T}{\Delta f} \quad (\text{avec } T_0 > T)$$

(voir fig. 4).

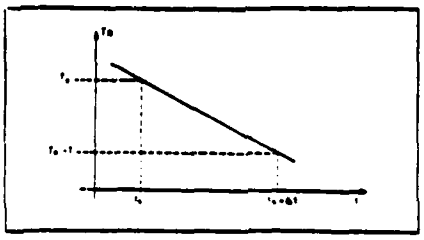


FIG. 4

telle ligne à retard est donnée par :

$$\varphi = -2\pi\int_0^f T_g\, df$$

$$\varphi = -2\pi\left[T_0+\frac{f_0 T}{\Delta f}\right]f+\pi\frac{T}{\Delta f}f^2$$

Et cette phase est bien l'opposé de $/\phi(f)$,

à un déphasage constant près (sans importance) et à un retard T_0 près (inévitable).

Un signal utile $S(t)$ traversant un tel filtre adapté donne à la sortie (à un retard T_0 près et à un déphasage près de la porteuse) un signal dont la transformée de Fourier est réelle, constante entre f_0 et $f_0+\Delta f$, et nulle de part et d'autre de f_0 et de $f_0+\Delta f$, c'est-à-dire un signal de fréquence porteuse $f_0+\Delta f/2$ et dont l'enveloppe a la forme indiquée à la figure 5, où l'on a représenté simultanément le signal $S(t)$ et le signal $S_1(t)$ correspondant obtenu à la sortie du filtre adapté. On comprend le nom de récepteur à compression d'impulsion donné à ce genre de filtre adapté : la « largeur » (à 3 dB) du signal comprimé étant égale à $1/\Delta f$, le rapport de compression est de $\frac{T}{1/\Delta f} = T\Delta f$

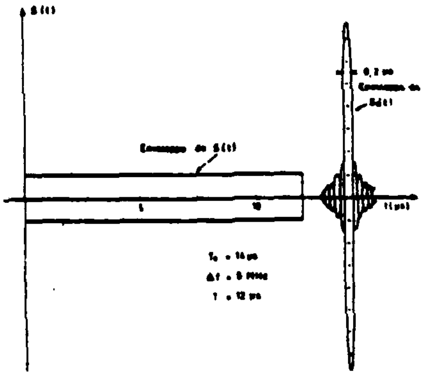


FIG. 5

On saisit physiquement le phénomène de compression en réalisant que lorsque le signal $S(t)$ entre dans la ligne à retard (LAR) la fréquence qui entre la première à l'instant 0 est la fréquence basse f_0, qui met un temps T_0 pour traverser. La fréquence f entre à l'instant $t=(f-f_0)\frac{T}{\Delta f}$ et elle met un temps $T_0-(f-f_0)\frac{T}{\Delta f}$ pour traverser, ce qui la fait ressortir à l'instant T_0 également. Ainsi donc, le signal $S(t)$

Figure 3-18

FULL LINE OF SYMBOLS Document 5

4. COMMONALITY

In this section, the commonality of a Mixed Mode machine with related machines is discussed. The related machines to be considered are:

(1) TELETEX machines

(2) Group 4 FACSIMILE Machines, without mixed mode capabilities

(3) Group 3 FACSIMILE machines

By commonality is meant the ability of a Mixed Mode machine to transmit messages to, or receive messages from, these other machines with a minimum of modification from its normal operation. Changes to the other machines considered are not permitted, since they will already be in the field.

A basic commonality has been designed into all the Mixed Mode techniques by the use of TELETEX code and the Modified READ II code proposed for Group 4 FACSIMILE machines. The Group 4 code differs from Group 3 code in that Group 4:

(1) uses $K = \infty$ instead of $K = 4$ for 7.7 lines/mm.

(2) deletes the EOL code for each line

(3) has no provision for stuffing bits to achieve a minimum line time.

(4) May allow wrap-around of run length codes over more than one scan line.

4.1 COMMONALITY WITH GROUP 3 FACSIMILE

Commonality with Group 3 machines for the various techniques will be directly related to the commonality with

Group 4 machines. It is assumed that converting messages between Group 3 and Group 4 machines will become common. Therefore there will be no further discussion of commonality with Group 3 machines, and anything said about commonality with Group 4 machines will also apply to Group 3 machines, plus the differences between Group 3 and Group 4 indicated above.

4.2 COMMONALITY WITH GROUP 4 FACSIMILE

Three of the techniques require almost no changes to be compatible with Group 4 transmissions. They are SYMBOL REMOVAL/LINE OF SYMBOLS, PARTIAL LINE OF SYMBOLS, and FULL LINE OF SYMBOLS. In each of these techniques, Group 4 transmissions can be received without any modifications whatsoever, and Group 4 transmissions can be produced by simply inhibiting all symbol recognitions. However, it may be necessary to inhibit information about the stored library to be used, which is true for all techniques.

In addition to the above, for SYMBOL REMOVAL/SCAN LINE and SYMBOL REMOVAL/FULL DOCUMENT, code bits that change mode must be deleted on transmission, and inserted on reception. For SYMBOL REMOVAL/SCAN LINE this is the single bit that precedes each scan line to indicate whether or not there are symbols on the scan line. For SYMBOL REMOVAL/FULL DOCUMENT, it is the single 8-bit code that indicates the change from symbols to graphics.

For EXTENDED TELETEX, a Group 4 transmission can be obtained by inhibiting all symbol recognitions, including blanks, which will force the entire line to be transmitted by graphics. In addition, the codes for the last symbol on the line and the graphics width would have to be deleted. For reception, the last symbol code and a graphics width code of 1,728 would have to be added before each scan line to convert the Group 4 transmission to what the Mixed Mode receiver expects.

4.3 COMMONALITY WITH TELETEX

The EXTENDED TELETEX technique requires almost no changes to be compatible with TELETEX transmissions. No change whatsoever is required to receive TELETEX transmission, except for adding the code that identifies the stored library to use. In transmission, the graphics mode must be inhibited, with space symbol codes being transmitted whenever material that cannot be recognized as symbols is encountered. Also the Carriage Return (CR) and Line Feed (LF) codes must be inserted for each line.

For all the techniques except EXTENDED TELETEX, in transmitting, the graphics mode must be inhibited, and a blank symbol used to replace each 20 pels of all-white or graphics pels. CR and LF symbols must be inserted at the end of each line (Approximately 33 scan lines). Corresponding changes must be made for reception, namely adding coding for approximately 33 all-white scan lines for each LF, and deleting the CR and LF codes.

In addition, for SYMBOL REMOVAL/LINE OF SYMBOLS, PARTIAL LINE OF SYMBOLS and FULL LINE OF SYMBOLS, the 12-bit (EOL) code that indicates a change from graphics to symbol mode must be deleted on transmission, and added on reception. For SYMBOL REMOVAL/FULL DOCUMENT the single 8-bit code that indicates the change from symbols to graphics must be deleted on transmission and added on reception.

5.0 COMPLEXITY OF IMPLEMENTATION

For the most part, there is not much to choose between the various techniques in terms of level of complexity of implementation. In all techniques, the most difficult problem is to recognize a group of black pels as a symbol, and to decide which of the symbols in the library it represents. Additional complexity is encountered by some of the techniques because of the amount of the image that must be stored, and due to the fact that in some techniques the recognized symbols must be organized into lines of symbols.

5.1 IMAGE STORAGE

In all of the techniques except SYMBOL REMOVAL/FULL DOCUMENT, the portion of the document that must be stored at any one time is an area equivalent to the height of the symbols by the full width of the document. Typically the height of an upper case character is 20 pels, but with large fonts and considering hang down characters, this could be as much as 32 pels. Since the width of the document is 1,728 pels, the total storage required is 32 x 1,728 = 55,296 bits. This much storage permits the recognition of symbols, removing them, and organizing them into lines.

For SYMBOL REMOVAL/FULL DOCUMENT, the symbols for the entire document are transmitted, and then the graphics by READ code. The entire document must first be scanned for symbols before any graphics are transmitted. Therefore either the entire document with symbols removed, or the READ code

for it, must be stored. The storage for the pels would require of course 2,376 x 1,728 = 4,105,728 bits. For Document 1, the READ code is only 31,828 bits, and for Document 5 it is only 37,894 bits (see section 2.2.2). While even more complex documents may be encountered, it seems that a storage of much less than the total number of pels in the document could be used. Another approach is to scan the document twice, once for symbol recognition and once to code the graphics portion. In any event, the amount of storage required for SYMBOL REMOVAL/FULL DOCUMENT appears to be at least several times that required for the other techniques, and possible up to 75 times as much.

5.2 ORGANIZATION INTO LINES

For the SYMBOL REMOVAL/SCAN LINE and SYMBOL REMOVAL/ FULL DOCUMENT techniques, each symbol that is recognized is incorporated into the transmission as the recognition occurs. For the remaining techniques, additional calculations are required to organize the symbols into lines of symbols that correspond to lines of print on the document. To do this, each symbol in the library must have stored with it the position of a baseline, which could be arbitrary, relative to the scan line on which the symbol is detected. The vertical position of each symbol must be adjusted according to this factor in order to bring all the symbols on a printed line into vertical alignment. In addition, small amounts of line skew must be accounted for by using the horizontal position

of each symbol. This can be done by calculating the linear regression of the horizontal and vertical symbol positions, to obtain the line skew. This skew is then used to correct the vertical position of each symbol.

In addition, for the EXTENDED TELETEX, PARTIAL LINE OF SYMBOLS, and FULL LINE OF SYMBOLS techniques, it is presumed that lines of symbols do not overlap, and so algorithms are required to decide which line of a set of overlapping lines is to be used for encoding symbols. *The simple equations of* Document 5 illustrate the problems that may be encountered in this area.

6.0 COMPARISON OF ALTERNATIVE TECHNIQUES

The compression estimates calculated in Section 3 for 200 lpi for all the segmentation techniques that were evaluated are summarized in Table 6-1. For comparison, the compression for normal Group 4 Modified READ ($k = \infty$, no EOL) is included. Similar compression estimates were also made for 300 lpi and 400 lpi, using the methods discussed in Section 2.3. The results of those calculations are shown in Tables 6-2 and 6-3.

Of the six alternative segmentation techniques evaluated for mixed-mode operation, none appears to have a clear-cut advantage over the other techniques. However, the SYMBOL REMOVAL/LINE OF SYMBOLS technique does appear to offer some advantages over the other techniques. First, the estimated compression is significantly (at least 10%) better than any other technique, even assuming a special CR/LF symbol for the techniques that could use it. The compression is better for both the simple and complex documents. Second, commonality with Group 3 and 4 Facsimile is as good as any, and is significantly better than some, especially EXTENDED TELETEX. Its commonality with TELETEX is as good as any technique except EXTENDED TELETEX. Third, in complexity it is as good as any as far as image storage is concerned, and is better than SYMBOL REMOVAL/FULL DOCUMENT. Fourth, as far as the requirement for organizing symbols into lines, it is inferior only to SYMBOL REMOVAL/SCAN LINE and SYMBOL REMOVAL/FULL DOCUMENT. In summary, SYMBOL REMOVAL/LINE OF SYMBOLS is superior in compression and commonality with Group 3 and 4 Facsimile, two very important factors, and is nearly as good

Technique	Document 1		Document 5	
	W/O CR/LF	WITH CR/LF	W/O CR/LF	WITH CR/LF
SYMBOL REMOVAL/SCAN LINE	79.6	-	55.3	-
SYMBOL REMOVAL/FULL DOCUMENT	79.5	-	53.2	-
SYMBOL REMOVAL/LINE OF SYMBOLS	101.8	-	74.3	-
EXTENDED TELETEX	44.7	83.4	35.2	43.4
PARTIAL LINE OF SYMBOLS	78.3	91.5	60.8	64.2
FULL LINE OF SYMBOLS	78.4	90.9	19.8	19.9
MODIFIED READ ($k = \infty$, no EOL)	27.9	-	15.8	-

TABLE 6-1

SUMMARY OF COMPRESSION ESTIMATES FOR 200 lpi

Technique	DOCUMENT 1		DOCUMENT 5	
	W/O CR/LF	WITH CR/LF	W/O CR/LF	WITH CR/LF
SYMBOL REMOVAL/SCAN LINE	131.4	-	94.5	-
SYMBOL REMOVAL/FULL DOCUMENT	131.2	-	94.9	-
SYMBOL REMOVAL/LINE OF SYMBOLS	164.2	-	124.4	-
EXTENDED TELETEX	85.6	141.8	67.1	80.5
PARTIAL LINE OF SYMBOLS	135.1	152.0	105.0	109.4
FULL LINE OF SYMBOLS	135.3	151.3	30.0	30.1
MODIFIED READ ($k = \infty$, no EOL)	41.8	-	23.7	-

TABLE 6-2

SUMMARY OF COMPRESSION ESTIMATES FOR 300 lpi

Technique	DOCUMENT 1		DOCUMENT 5	
	W/O CR/LF	WITH CR/LF	W/O CR/LF	WITH CR/LF
SYMBOL REMOVAL/SCAN LINE	207.3	-	137.4	-
SYMBOL REMOVAL/FULL DOCUMENT	185.2	-	130.1	-
SYMBOL REMOVAL/LINE OF SYMBOLS	227.6	-	176.1	-
EXTENDED TELETEX	132.7	202.7	103.4	120.6
PARTIAL LINE OF SYMBOLS	194.8	214.2	151.5	156.7
FULL LINE OF SYMBOLS	195.0	213.4	40.9	41.0
MODIFIED READ ($k = \infty$ no EOL)	55.8	-	31.6	-

TABLE 6-3

SUMMARY OF COMPRESSION ESTIMATES FOR 400 lpi

as any segmentation technique as far as the other factors are concerned.

PARTIAL LINE OF SYMBOLS appears to be the second choice. It offers better overall compression than any technique except SYMBOL REMOVAL/LINE OF SYMBOLS. FULL LINE OF SYMBOLS has equal compression for the simple document (DOCUMENT 1) but it is poor for the complex document (DOCUMENT 5). Also PARTIAL LINE OF SYMBOLS is as good as SYMBOL REMOVAL/LINE OF SYMBOLS as far as commonality with Group 3 and 4 facsimile and image storage complexity is concerned. In organizing lines of symbols, additional logic is required to prevent lines from overlapping vertically, which is not required for SYMBOL REMOVAL/LINE OF SYMBOLS.

SYMBOL REMOVAL/SCAN LINE, SYMBOL REMOVAL/FULL DOCUMENT, and EXTENDED TELETEX (with CR/LF symbol) give about equal compression for both simple and complex documents. Of these techniques, SYMBOL REMOVAL/SCAN LINE has an edge over SYMBOL REMOVAL/FULL DOCUMENT because the storage requirement is less. Both techniques are superior to EXTENDED TELETEX because of the latter's requirement for a complex algorithm to organize symbols into lines, and because of its poor commonality with Group 3 and 4 Facsimile, an important consideration. Also EXTENDED TELETEX was presented as a technique for generating a mixed-mode message by a computer, not a method for scanning a document and segmenting it into graphics and text.

Finally, FULL LINE OF SYMBOLS can give poor compression

on complex documents, and requires a complex algorithm to organize symbols into lines.

Table 6-4 summarizes the subjective evaluations given to each segmentation technique for each of the topics considered in the evaluation.

TABLE 6-4

RELATIVE EVALUATION OF ALTERNATIVE TECHNIQUES

Technique	COMMONALITY		COMPLEXITY		COMPRESSION
	GROUP 3/4	TELETEX	IMAGE STORAGE	LINE ORGANIZATION	
SYMBOL REMOVAL/SCAN LINE	Good	Good	Good	Excellent	Fair
SYMBOL REMOVAL/FULL DOCUMENT	Good	Good	Fair	Excellent	Fair
SYMBOL REMOVAL/LINE OF SYMBOLS	Excellent	Good	Good	Good	Excellent
EXTENDED TELETEX	Fair	Excellent	Good	Fair	Fair
PARTIAL LINE OF SYMBOLS	Excellent	Good	Good	Fair	Good
FULL LINE OF SYMBOLS	Excellent	Good	Good	Fair	Poor for Complex Document
MODIFIED READ	Perfect for 4 Excellent for 3	Not Possible	Excellent	Excellent	Poor
TELETEX	Not Possible	Perfect	Excellent	Poor	Fair

7. RECOMMENDATIONS

The mixed-mode segmentation technique, SYMBOL REMOVAL/ LINE OF SYMBOLS appears to be the preferred approach and should be pursued further.

Further study is needed in the following areas.

1) Compression estimates should be made on a wider range of documents.

2) Compression should be measured for a few selected techniques using computer simulation.

3) Studies should be made of more efficient coding techniques, such as a shorter code for graphics width.

4) Studies should be made of techniques for organizing symbol detections into lines of symbols.

Appendix A

International Digital Facsimile Coding Standards

International Digital Facsimile Coding Standards

ROY HUNTER AND A. HARRY ROBINSON

Invited Paper

Abstract—Recently Study Group XIV of CCITT has drafted a new Recommendation (T.4) with the aim of achieving compatibility between digital facsimile apparatus connected to general switched telephone networks. A one-dimensional coding scheme is used in which run lengths are encoded using a modified Huffman code. This allows typical A4 size documents in the form of black and white images scanned at normal resolution (3.85 lines/mm, 1728 pels/line) to be transmitted in an average time of about a minute at a rate of 4800 bit/s. The Recommendation also includes a two-dimensional code, known as the modified relative element address designate (READ) code, which is in the form of an optional extension to the one-dimensional code. This extension allows typical documents scanned at high (twice normal) resolution (with every fourth line one dimensionally coded) to be transmitted in an average time of about 75 s at 4800 bit/s. This paper describes the coding schemes in detail and discusses the factors which led to their choice. In addition, this paper assesses the performance of the codes, particularly in relation to their compression efficiency and vulnerability to transmission errors, making use of 8 CCITT reference documents.

I. Introduction

STUDY GROUP XIV of the CCITT currently defines 2 Recommendations (T.2 and T.3) for the transmission of documents by facsimile over the general switched telephone network. These refer to Groups 1 and 2 type facsimile apparatus and allow A4 size documents scanned at 3.85 lines/mm to be transmitted in 6 and 3 min, respectively. Faster transmission cannot be readily obtained using analogue techniques because of the restricted bandwidth of voiceband telephone circuits. However, many documents that are likely to be transmitted, such as business letters, forms, and diagrams, can be satisfactory reproduced when quantized and transmitted in the form of two tones, i.e., black and white. Hence, a new Recommendation (T.4) has been drafted for Group 3 type apparatus which relies on digital transmission techniques. The aim of the standard is to enable two-tone A4 documents scanned at a (normal) resolution of 3.85 lines/mm and sampled at 1728 samples/line to be transmitted at 4800 bit/s in an average time of about 1 min over the general switched telephone network. It is hoped that the draft Recommendation T.4 will be ratified by CCITT in late 1980.

In order to achieve this transmission time, the amount of digital data representing the document image must be reduced by source coding. At the end of 1977, SGXIV of CCITT agreed that Group 3 equipment should use a one-dimensional run-length coding scheme and a modified Huffman code. A one-dimensional code was chosen as a suitable compromise between obtaining a high compression efficiency while minimizing the susceptibility to transmission errors and keeping the implementation costs to a low level. Its freedom from patent restrictions was also an important consideration. In 1979, an optional two-dimensional coding scheme, in the form of an extension to the one-dimensional coding scheme, was added to the Recommendation T.4 for Group 3 apparatus. The two-dimensional scheme allows greater compression efficiency to be obtained for many documents, particularly when they are scanned at twice normal vertical resolution. The additional factors considered when chosing this code were compatibility with the one-dimensional code and possible future extensions to other codes.

In this paper, we describe both of the coding schemes included in the Recommendation T.4, as well as the factors which influenced the selection of these schemes and assess their performances. Results are given on coding statistics and compression efficiencies, as measured by computer simulation on 8 CCITT reference documents (Figs. 1 and 2). We have used 2 test procedures for assessing the error susceptibility of the schemes. The results provide useful information about the likely extent of damage to documents caused by transmission errors and of the effectiveness of several methods which may be used to reduce the subjective effects of such damage.

Many companies and national telecommunications administrations undertook similar evaluation studies, and it was only through widespread collaboration and agreement under the auspices of the CCITT that it was possible to draft a satisfactory Recommendation T.4.

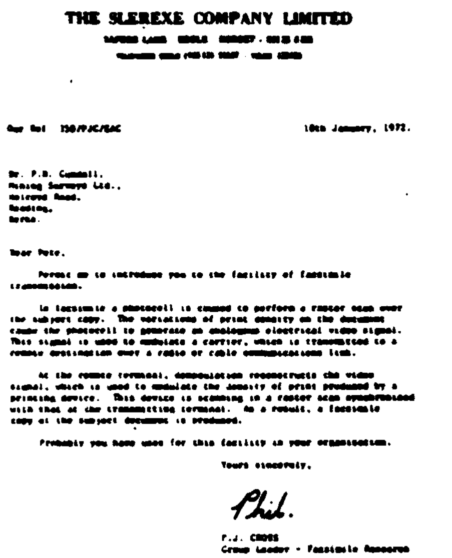
Section II discusses some of the factors considered in the choice and design of the one-dimensional coding scheme. Section III outlines the standards for Group 3 apparatus and specifies parameters such as document size, resolution, scanning rate, and modulation methods. Section IV describes the Group 3 one-dimensional coding standard and Section V summarizes its performance, particularly in respect to its susceptibility to transmission errors. Section VI outlines the criteria used to select the two-dimensional coding scheme described in Section VII. In Section VIII, we give the compression efficiency results for this code and discuss briefly some error susceptibility measurements.

II. Criteria for Selecting a One-Dimensional Group 3 Code

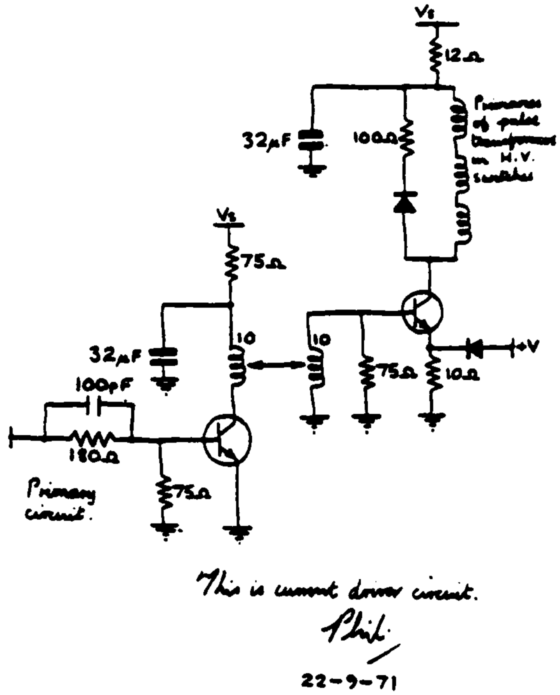
As stated in the Introduction, the main purpose of the Group 3 standard is to allow typical A4 documents to be

Manuscript received February 15, 1980; revised March 10, 1980.
The authors are with the British Post Office Centre, Martlesham Heath, Ipswich IP5 7SE, England.

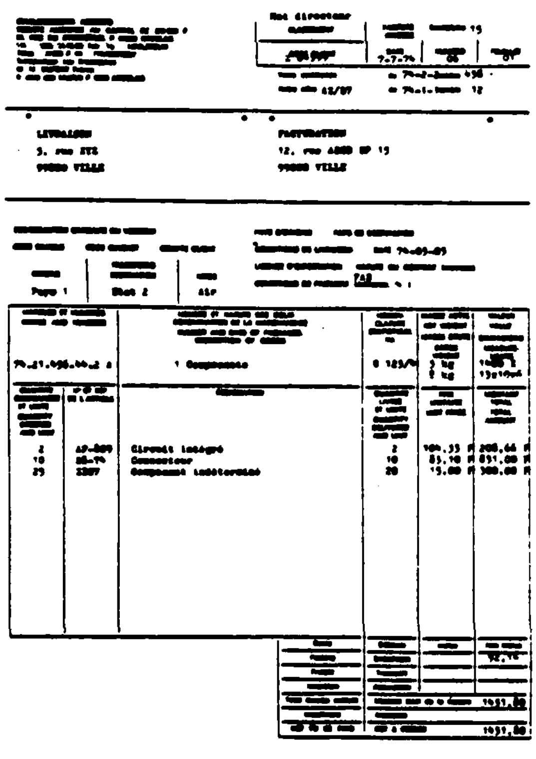
0018-9219/80/0700-0854\$00.75 © 1980 IEEE



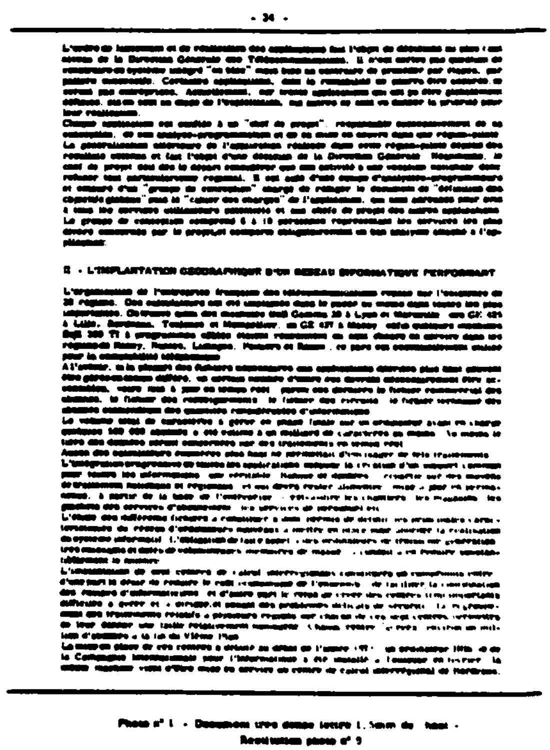
Document 1



Document 2

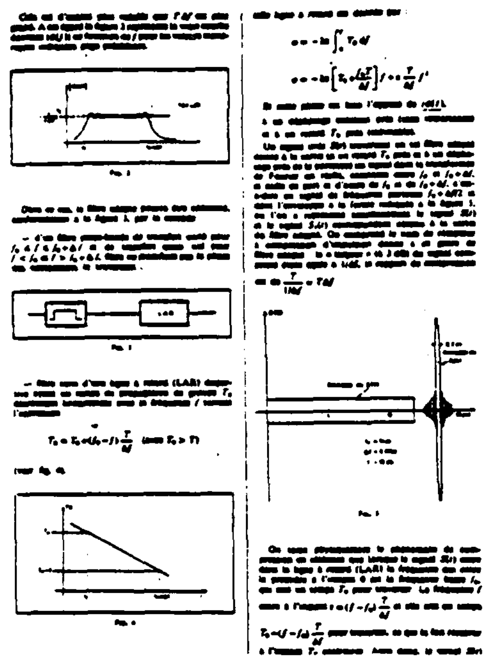


Document 3

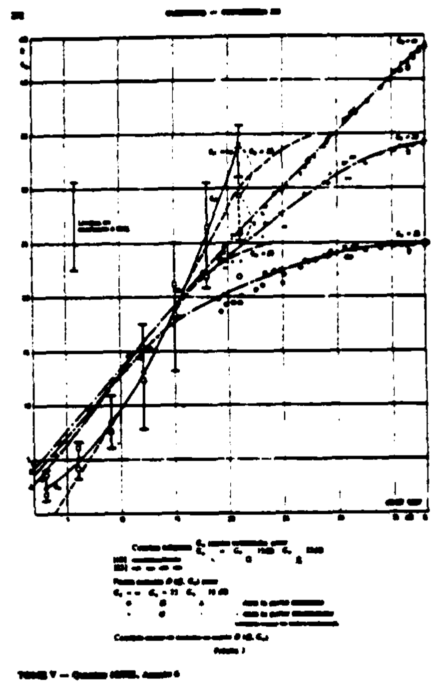


Document 4

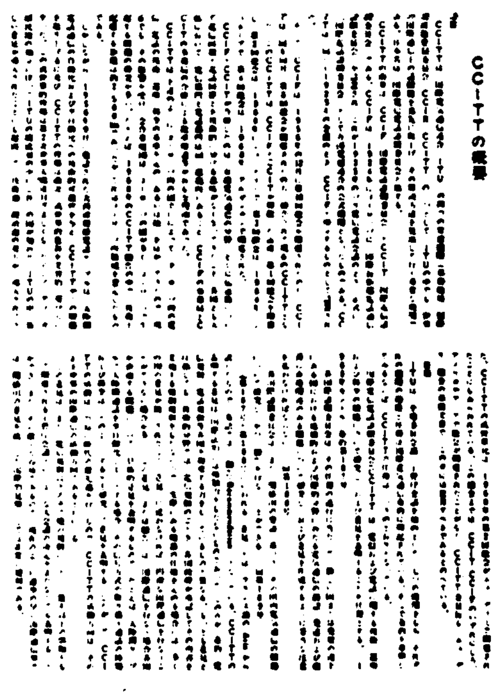
Fig. 1. CCITT test documents.



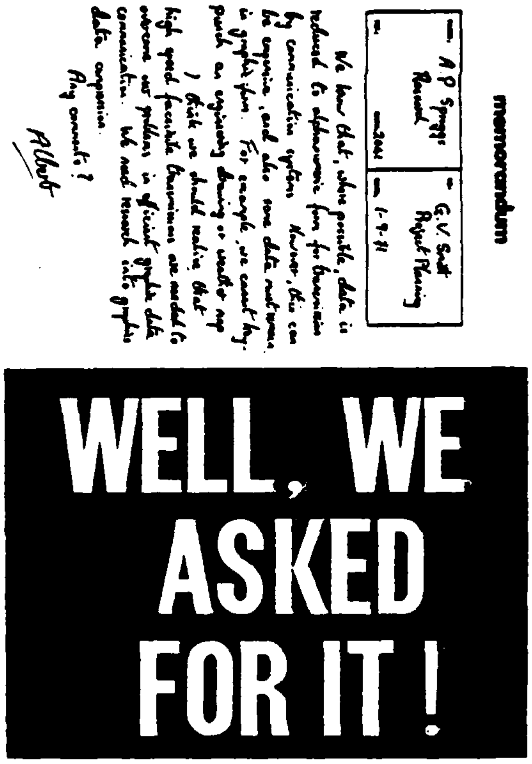
Document 5



Document 6



Document 7



Document 8

Fig. 2. CCITT test documents.

transmitted digitally over telephone networks in an average time of 1 min. The digital image representing a document is obtained by scanning it and quantizing each sample or pel into one of two logical levels, representing black and white. At a normal Group 3 scanning density of 3.85 lines/mm (about 1188 lines on an A4 page) with each line divided into 1728 pels, the amount of data generated would require, in the absence of source coding, a transmission time of just over 7 min at a transmission rate of 4800 bit/s. However, due to the strong statistical dependencies between pels, the transmission times for most documents can be considerably reduced with a suitable source coding method.

The relative importance of factors such as compression efficiency, error susceptibility, and complexity of implementation depend upon each facsimile application. A high compression efficiency is particularly important for high volume usage. In other applications, the need is for reliable equipment providing acceptable copies over national or international telephone networks at reasonable terminal costs. In addition, the mechanical limitation of some equipment must be taken into account, and machines with a wide range of facilities must be able to interwork with more basic equipment.

Based on these criteria, a one-dimensional run-length coding scheme using a modified Huffman code was chosen as a basic Group 3 standard. Run-length codes have been widely employed in data reduction systems and are easy to implement. In most cases, Huffman codes offer high compression efficiencies and the use of a "modified" code simplifies implementation. Damage caused by errors is kept to acceptable levels by using a single line coding scheme and by transmitting a robust line synchronization codeword. Other codes were investigated but were found to be equally susceptible to errors and generally performed less efficiently.

A. Compression Efficiency

Numerous source coding techniques have been applied to digital facsimile signals in order to reduce the amount of data required to transmit them. Data reduction is achieved by using a code which exploits the statistical dependencies between pels. The facsimile signal is characterized by a source alphabet for which source symbols have low statistical dependency. A code table is used which provides a statistically optimum match between the source alphabet and the codewords. Codes which make use of only the horizontal dependencies between pels on the same scan line are usually classed as one-dimensional codes, while two-dimensional codes attempt to obtain greater efficiency by exploiting both horizontal and vertical dependencies. Shannon's theory of communication [4] indicates the amount of data reduction that can be achieved by statistical source coding methods, but the actual statistical dependencies between pels can only be determined by experiment.

A useful model of a digital image has been proposed by Capon [5]. Each scan line is regarded as a first-order Markov chain in which the color of each pel X_i is dependant only on the color of the previous pel X_{i-1}. The average amount of information per pel is given by the entropy H_{pel}, where

$$H_{pel} = \sum_{X_{i-1}} \sum_{X_i} P(X_{i-1}, X_i) \cdot \log P(X_i/X_{i-1}) \qquad (1)$$

where the summation is taken over all possible combinations of 2 adjacent pels.

This model leads to run-length coding techniques in which the digital image is regarded as a sequence of alternating independent runs of black and while pels. Two source alphabets are formed, one consisting of all the black run-length values and the other containing the set of white runs. The average white run-length value is given by

$$\bar{r}_w = \sum_{r=0}^{n} r \cdot P_w(r) \qquad (2)$$

where $P_w(r)$ is the probability of a white run of length r and n is the largest value of r. The average amount of information in bits for each white run is given by the entropy

$$H_w = -\sum_{r=0}^{n} P_w(r) \cdot \log_2 P_w(r). \qquad (3)$$

Similar equations can be written for the average black run-length value $\bar{r}_b$ and the entropy of the black runs H_b.

The entropy per pel H_{pel} and the maximum theoretical compression factor Q_{max} for a given set of run-length values are given by

$$Q_{max} = \frac{1}{H_{pel}} = \frac{\bar{r}_w + \bar{r}_b}{H_w + H_b}. \qquad (4)$$

H_{pel} in (4) is usually higher than H_{pel} for the Capon model given in (1) and indicates that the run-length coding model includes some of the higher order dependencies between successive pels of the same color. There are, of course, dependencies between adjacent runs on the same line which have been ignored in the above analysis. However, these dependencies have been found to be small [6], providing about a 10 percent decrease in the value of H_{pel}.

In the above analysis, the white and black runs have been placed in separate source alphabets. If the black and white run-length distributions are combined into one source alphabet with entropy H_c, then $H_c \geq \frac{1}{2}(H_w + H_b)$ with equality occurring only when $P_w(r) = P_b(r)$ for all values of r. Measurements on typical documents have shown that Q_{max} is increased on average by about 25 percent if separate distributions are used instead of a single distribution. It is generally agreed that the advantage of this increase in the available efficiency outweighs the disadvantage of having to use two separate code tables.

Many different codes have been designed for use in run-length coding. However, it can be shown that Huffman's procedure [7] is the optimum method of constructing a uniquely decodable and instantaneous code which has, for a given independent source alphabet, the smallest average codeword length. Such a code is usually called a compact code. For example, the average number of coded bits per run for the compact code designed for the white run lengths will be in the range

$$H_w \leq \sum P_w(r) \cdot n_w(r) < H_w + 1 \qquad (5)$$

where $n_w(r)$ is the length of the codeword representing the white run-length r. The code is instantaneous since it is a prefix code for which no codeword can appear as the beginning of any other codeword in the same code table. Thus codewords can be decoded as soon as they are received. Also the code is exhaustive so that any sequence of binary digits can be decoded.

The Huffman code tables used in the Group 3 standard were designed from the run-length statistics averaged over many typical documents. Two Huffman codes would result in a large number of codewords (1728 in each) and hence a "modified" code was designed in which every run length greater than 63 is broken into 2 run lengths, namely a make-up run length having a value $N \times 64$ (where N is an integer) and a terminating run length having a value between 0 and 63. This reduces the number of codewords required and simplifies implementation. The extra run lengths required lead to modified source statistics but the reduction in efficiency is small. Furthermore, measurements have shown that the modified Huffman code tables used in the Group 3 standard are reasonably insensitive to considerable changes in the source statistics and most documents can be transmitted with high efficiency.

B. Error Susceptibility

An acceptable digital facsimile system for use on the general switched telephone network must include some method of dealing with transmission errors and of limiting their effects. Three methods have been considered.

1) restriction of the damage caused by errors to as small an area on the document as possible;
2) detection of errors and retransmission of blocks of data in error using an automatic repeat request (ARQ) system;
3) detection of errors and their correction at the receiver using a forward-acting error correcting (FEC) code.

The Group 3 coding system has been designed so that the damage caused by a transmission error is confined, where possible, to the scan line in which the error occurs. This is achieved by transmitting a special synchronizing sequence called the end-of-line (EOL) codeword at the end of each coded line. This codeword is unique since it consists of a sequence of digits which cannot occur naturally anywhere in a scan line of coded data (see Section IV-C). It can therefore be easily recognized and although a coded line is damaged by a transmission disturbance, all subsequent lines can be correctly received and decoded. This method has a further advantage; since each correctly decoded line represents exactly 1728 pels, a damaged line containing a different number of decoded pels can be detected. The receiver may then be able to reduce the subjective effect of the damage on the document by employing one of the error concealment processes described in Section V-D.

ARQ and FEC methods were considered but neither have been incorporated in the Group 3 standard since they are complex, add extra cost to the equipment, and increase the transmission times. Furthermore, there is insufficient evidence to show whether either of these methods would be suitable on general telephone networks and manufacturers' field experience indicates that acceptable document quality can be obtained for most facsimile calls without the use of error correction methods. However, studies indicate that error control techniques may be applicable to future facsimile equipment.

ARQ systems have the advantage of being very reliable and insensitive to changes in the channel error rate. Systems can be designed such that the probability of an undetected error in a block (of say 2048 bits) in less than 1 in 10^8. However, they lead to a reduction in the effective transmission rate—the reduction factor is often referred to as the throughput. For example, following Burton's [8] analysis for a 4.8 kbit/s V27 ter modem, a "stop-and-wait" ARQ system has a throughput as low as 0.85 on a national terrestial circuit and as low as 0.27 on a single satellite circuit. For a continuous ARQ system, which requires an asymmetrical full duplex modem with a backward channel rate of 75 or 150 bit/s, the corresponding throughputs are as low as 0.89 and 0.84, respectively.

FEC codes have higher effective transmission rates but must be designed for the errors experienced on practical circuits and can be very sensitive to changes in those error patterns. Recently, an implementation of a Reed–Solomon code on a microprocessor has been demonstrated [9]. This system is capable of correcting a single burst error spread over up to 17 bits in a block of 255 bytes, and has a throughput of greater than 0.97. However, it is difficult to assess the performance of this system since, in general, the error characteristics of most switched telephone networks are not sufficiently well known.

C. Complexity of Implementation

In the past, Huffman codes have not been used in document facsimile systems because they were generally regarded as being difficult to implement. For example, Huffman decoders usually relied on simple table-lookup or tree-follower methods which require a large amount of storage and are slow in operation. It was also thought that the codes did not recover quickly from transmission errors. Instead of a number of other codes, comma or block codes [3], [10], [11], [12] were proposed for facsimile use. Although they were easier to implement, they usually gave somewhat lower compression factors and were more sensitive to changes in document statistics. Recently, methods of decoding Huffman codes have been devised which are fast and require a modest amount of storage [13], [14]. Furthermore, after the modified Huffman code had been proposed for the Group 3 standard, its recovery properties were studied in detail. From this investigation it was concluded that the code performed satisfactorily in respect of its susceptibility to errors.

One code in particular, the intermediate ternary code (ITC) [3], [15] was considered as a possible alternative code for Group 3 equipment since it was thought that in the presence of errors, it might perform better than a Huffman code. In the ITC scheme, the binary numbers representing the values of the run lengths are converted to a ternary state which distinguishes between black and white runs. Ternary pentades are then converted into binary octades which form the coded data. In its basic form, this code does not require a code table and therefore the implementation is simple. Also, since the codewords are all 8 bits long, there can be no loss of codeword synchronization. However, a comparison of the effects of errors on the ITC code with the effects on the modified Huffman code [16], indicated that there was very little practical difference between the codes. In addition, the ITC code produced slightly lower compression factors than the modified Huffman code.

III. Elements of the Group 3 Recommendations

The specifications needed to provide for interworking of Group 3 facsimile equipment over the general switched telephone network are given in CCITT Recommendations T.4 and T.30. Recommendation T.4 contains standards concerning

the following parameters: document size, resolution, scanning rate, source coding methods, and modulation method. Associated with these are a number of options which enable facsimile equipment to communicate in alternative modes, e.g., a higher vertical resolution is provided so that higher quality copies can be obtained. Two facsimile machines may communicate using any of the options by mutual agreement, otherwise they must use the appropriate recommended standard.

Recommendation T.30, which is not covered by this paper, specifies the digital signals and procedures used by Group 3 apparatus for: call setup, premessage procedures for identifying and selecting the required facilities, message transmission, postmessage procedure, and call release. The following subsections summarize the parameters defined in T.4.

A. Dimensions of Apparatus

1) Facsimile machines should be able to accept A4 size documents. As an option, documents up to A3 in size may be transmitted with the same resolution.

2) The normal vertical resolution is 3.85 lines/mm. A high vertical resolution of 7.7 lines/mm is available as an option.

3) Each scan line on an A4 document is divided into 1728 black and white pels. The number of pels may be optionally increased to about 2600 to allow documents up to A3 size to be transmitted at the same resolution.

4) The scanning line length on an A4 document is 215 mm. Other line lengths may be used provided that the vertical resolution is adjusted to maintain the correct picture proportions.

The normal vertical resolution is chosen to be the same as that used in Groups 1 and 2. A higher resolution is included to allow higher quality copies to be obtained. A horizontal resolution which is nearly twice that of the normal vertical resolution is required to ensure that staircase effects on vertical black and white edges due to the sampling and quantizing processes do not impair legibility.

B. Minimum Scan Line Times and Message Format

These are specified so that transmitters and receivers can keep in step on a line-by-line basis and to allow for mechanical limitations of some machines. Also, some receivers operate at normal resolution by printing each scan line twice at high resolution. The recommended standard minimum scan line times (MSLT) is 20 ms (equivalent to a minimum of 96 coded bits at a transmission rate of 4800 bit/s) and there are options of 10 ms (48 bits), 5 ms (24 bits) and 0 ms (i.e., no restriction). Any machine offering an option must be able to operate at all longer MSLT's down to 20 ms. The recommendation also includes a 40 ms option. The coding procedure includes a method of adding varying length strings of "fill" bits to those coded lines containing fewer than the required number of bits. Fill bits are easily recognized by the receiver and are discarded.

Fig. 3 shows the format of the data for several coded lines. The end of document transmission is indicated by 6 consecutive EOL codewords which form the return to control (RTC) signal.

C. Modulation and Demodulation Methods

When operating on the general switched telephone network, it is recommended that Group 3 equipment should use data rates of 4800 bit/s and 2400 bit/s and the modulation, scrambler, equalization, and timing signals defined in CCITT Recommendation V27 ter. However, where higher speeds of operation are possible, it has been agreed that Group 3 equipment may operate optionally at 9600 bit/s, and 7200 bit/s using the modulation, scrambler, equalization and timing signals defined in CCITT Recommendation V29. Some PTT Administrations point out, however, that it may not be possible to guarantee the service at a data signalling rate higher than 2400 bit/s.

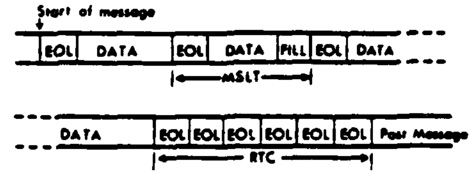


Fig. 3. Message format.

IV. One-Dimensional Coding Scheme

This code was first suggested by the Plessey Company in 1976. Later a revised version of the code was proposed jointly by a number of British and American companies under the auspices of the British Facsimile Industries Compatibility Committee (BFICC) and the Electronic Industries Association (EIA) [17]. It is this version of the code that was eventually accepted by SGXIV of the CCITT. The extended code table described in Section IV-B is due to a proposal made by the British Post Office.

A. The Coding Scheme

Each scan line is regarded as a sequence of alternating black and white lines. All lines are assumed to begin with a white run to ensure that the receiver maintains color synchronization; if the first actual run on a line is black, then a white run of zero length is transmitted at the beginning of the line.

Separate code tables are used to represent the black and white runs and these are given in Table I. Each code table can represent a run-length value up to the maximum length of one scan line (1728 pels) and contains two types of codewords: terminating codewords (TC) and make-up codewords (MUC). Runs between 0 and 63 pels are transmitted using a single terminating codeword. Runs between 64 and 1728 are transmitted by a MUC followed by a TC. The MUC represents a run-length value of $64 \times N$ (where N is an integer between 1 and 27) which is equal to, or shorter than, the value of the run to be transmitted. The following TC specifies the difference between the MUC and the actual value of the run to be transmitted.

The coding of each scan line continues until all runs on the line (i.e., a total of 1728 pels) have been transmitted. Each coded line is followed by the EOL codeword. As stated in Section II-B, the EOL codeword is a unique sequence which cannot occur within a valid line of coded data. It can be detected irrespective of the way in which the decoder breaks up the coded line into codewords. Thus, if a transmission error corrupts some of the coded scan line data, then the error cannot prevent the EOL from being detected.

If the number of coded bits in a line is fewer than a certain agreed minimum (Section III-B), then "fill" bits consisting of varying length strings of "0"'s are inserted between the line of coded data and the EOL codeword.

TABLE I
MODIFIED HUFFMAN CODE TABLE

Run-length	white runs	black runs
	- Terminating Codewords -	
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111
13	000011	00000100
14	110100	00000111
15	110101	000011000
16	101010	0000010111
17	101011	0000011000
18	0100111	0000001000
19	0001100	00001100111
20	0001000	00001101000
21	0010111	00001101100
22	0000011	00000110111
23	0000100	00000101000
24	0101000	00000010111
25	0101011	00000011000
26	0010011	000011001010
27	0100100	000011001011
28	0011000	000011001100
29	00000010	000011001101
30	00000011	000001101000
31	00011010	000001101001
32	00011011	000001101010
33	00010010	000001101011
34	00010011	000011010010
35	00010100	000011010011
36	00010101	000011010100
37	00010110	000011010101
38	00010111	000011010110
39	00101000	000011010111
40	00101001	000001101100
41	00101010	000001101101
42	00101011	000011011010
43	00101100	000011011011
44	00101101	000001010100
45	00000100	000001010101
46	00000101	000001010110
47	00001010	000001010111
48	00001011	000001100100
49	01010010	000001100101
50	01010011	000001010010
51	01010100	000001010011
52	01010101	000000100100
53	00100100	000000110111
54	00100101	000000111000
55	01011000	000000100111
56	01011001	000000101000
57	01011010	000001011000
58	01011011	000001011001
59	01001010	000000101011
60	01001011	000000101100
61	00110010	000001011010
62	00110011	000001100110
63	00110100	000001100111
	- Make-up Codewords -	
64	11011	0000001111
128	10010	000011001000
192	010111	000011001001
256	0110111	000001011011
320	00110110	000000110011
384	00110111	000000110100
448	01100100	000000110101
512	01100101	0000001101100
576	01101000	0000001101101
640	01100111	0000001001010
704	011001100	0000001001011
768	011001101	0000001001100
832	011010010	0000001001101
896	011010011	0000001110010
960	011010100	0000001110011
1024	011010101	0000001110100
1088	011010110	0000001110101
1152	011010111	0000001110110
1216	011011000	0000001110111
1280	011011001	0000001010010
1344	011011010	0000001010011
1408	011011011	0000001010100
1472	010011000	0000001010101
1536	010011001	0000001011010
1600	010011010	0000001011011
1664	011000	0000001100100
1728	010011011	0000001100101
EOL	000000000001	000000000001

B. The Extended Code Tables

The Group 3 standard provides an optional extension to the coding scheme allowing machines to transmit larger paper widths up to A3 in size which require nearly 2600 pels/line. This option is provided by 2 extended code tables formed by adding 13 extra MUC listed in Table II to each of the basic code tables given in Table I. The construction of the extra codewords is described in Section IV-C. The use of the extended code table is signalled in the Recommendation T.30 control procedures.

TABLE II
EXTENDED MODIFIED HUFFMAN CODE TABLE

Run Length (Black or White)	Make-up Codeword
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

C. Construction of the Modified Huffman Code Tables

The properties of the EOL codeword can be further understood by considering the construction of the modified Huffman code tables. Each code table was initially designed according to Huffman's procedure and to contain the codeword "0000000" (7 × "0") which was designated to signal the end of a scan line. Redundant bits were then added to the codeword 7 × "0" to form the codeword 10 × "0" + "1". By examining Table I, it can be seen that no codeword ends in a sequence of more than 3 "0"'s or begins with a sequence of "0"'s larger than 6 and therefore 10 × "0" + "1" forms a unique sequence which cannot be produced by a concatenation of codewords. The final "1" of the EOL is included to indicate the start of the next coded line, since fill bits may extend the sequence beyond 10 × "0".

The extended black and white code tables were formed using the codeword 7 × "0" as the prefix for the 13 extra MUC. The 7 × "0" codeword originally designated to signal the end of a scan line now needs to be increased to 8 × "0". Redundant bits are then added to this codeword to form the EOL codeword 11 × "0" + "1" which is unique using either the basic or extended code tables. This process can be carried out without altering any of the other codewords in Table I. The same 13 extra codewords can be added to each code table without a loss in efficiency since long runs occur very infrequently.

V. PERFORMANCE OF THE ONE-DIMENSIONAL CODING SCHEME

The performance of the one-dimensional coding scheme was assessed by computer simulation. Measurements were made on run-length data obtained for the 8 CCITT A4 reference documents which were recorded on magnetic tape by CCITT on behalf of the French Administration of Posts and Telecommunications. Each document was scanned at high resolution (7.7 lines/mm) and contained 2376 lines. Measurements made at normal resolution (3.85 lines/mm) used the 1188 odd numbered lines.

Two main types of simulation were performed to assess the error sensitivity of the one-dimensional coding scheme. Firstly, coded data obtained using the one-dimensional coding scheme was subjected to single random errors in order to study the resynchronization properties of the modified Huffman code (Section V-B). This method was originally designed to compare the resynchronization properties of various variable length, comma-free codes. Secon.., , error susceptibility measurements were made on a number of documents using

TABLE III
ENTROPY AND Q_{max} VALUES FOR THE BASIC RUN-LENGTH DISTRIBUTION

Doc	$\bar{r}_w$	$\bar{r}_b$	H_w	H_b	Q_{max}
1	156.3	6.793	5.451	3.992	18.03
2	257.1	14.31	8.163	4.513	21.41
3	89.81	8.515	5.688	3.572	10.62
4	39.00	5.676	4.698	3.124	5.712
5	79.16	6.986	5.740	3.388	9.500
6	138.5	8.038	6.204	3.641	14.89
7	45.32	4.442	5.896	3.068	5.553
8	85.68	70.87	6.862	5.761	12.40

TABLE IV
ENTROPY AND Q_{max} VALUES FOR THE MODIFIED RUN-LENGTH DISTRUBITON

Doc	$\bar{r}_w$	$\bar{r}_b$	H_w	H_b	Q_{max}	C
1	134.6	6.790	5.230	3.998	16.02	15.16
2	167.9	14.02	5.909	4.457	17.41	16.67
3	72.50	8.468	5.189	3.587	9.112	8.390
4	36.38	5.673	4.576	3.126	5.461	4.911
5	66.41	6.966	5.280	3.339	8.513	7.987
6	90.65	8.002	5.063	3.651	11.32	10.78
7	39.07	4.442	5.380	3.068	5.188	4.990
8	64.30	60.56	4.427	5.310	11.52	8.665

real error patterns obtained from actual telephone lines. The error patterns, which were recorded on magnetic tape, were produced by the University of Hannover by transmitting pseudorandom patterns over telephone lines using a 4.8 kbit/s V27 ter modem and by comparing them with the received patterns. The magnetic tape contained error patterns for 4 telephone lines and the first of these were used to produce the results given later in this paper.

A. Compression Factors

Table III lists the average run lengths, entropies, and maximum theoretical compression factors (Q_{max}) for the run-length statistics obtained from the 8 CCITT documents scanned at high resolution. Table IV gives the corresponding values for the modified statistics obtained by breaking up runs greater than 63 into 2 runs as required by the modified Huffman code and indicates that Q_{max} has been reduced by about 14 percent. To indicate that the coding scheme performs efficiently for a wide range of documents, Table IV also includes the actual compression factors C which were calculated using the modified Huffman code but excluding EOL codewords and fill bits.

Table V, which lists the number of coded bits obtained for the documents scanned at normal (low) and high resolution,

TABLE V
NUMBER OF CODED BITS OBTAINED FOR ONE-DIMENSIONAL GROUP 3 CODING

	LOW RESOLUTION		HIGH RESOLUTION	
DOCUMENT	0 msecs	20 msecs	0 msecs	20 msecs
1	149834	201200	299311	402419
2	137252	161897	274858	324144
3	260247	277607	520196	554970
4	432219	460484	864524	921059
5	273164	290525	546460	581281
6	204516	225781	409290	451894
7	426053	443779	851286	884623
8	251171	263161	502331	526444
AVERAGE	266807	290439	533532	580854

indicates that an increase in the MSLT parameter can substantially increase the number of bits required to code a document. At low resolution, the average transmission times for the eight documents using a transmission rate of 4800 bit/s are: 56.6 s for a MSLT of 0 ms, and 60.5 s for a MSLT of 20 ms. At high resolution, the corresponding times are 111 and 121 s, respectively.

Care should be taken when comparing the compression values listed in this paper with those obtained by other experimenters. Even different scanners operating at the same resolution and scanning the same documents can give significantly different statistics and compression factors. For example, another version of the 8 CCITT documents produced compression factors which were on average 13 percent lower than the values quoted in this paper.

B. Analysis and Measurement of the Error Effects on a Scan Line of Coded Data

When a coded scan line of facsimile data is disturbed by a transmission error, a number of related effects are produced. At least one of the codewords will be corrupted and incorrectly decoded. Also, the length of the codeword recognized by the decoder may be different from that generated by the coder so that the coder and decoder will become out of step. Decoding will continue since the codes are exhaustive (ignoring for a moment the redundancy associated with the EOL codewords) but codewords recognized by the decoder may not be the same as those formed by the coder. Eventually resynchronization will occur, this being a property of most comma-free codes [18], [19], after which the data will be correctly decoded. If resynchronization does not occur naturally along the coded line then the specially constructed EOL codeword forces resynchronization at the end of the coded line.

Fig. 4 shows an example of the effect of a single error on part of a typical coded data stream. The resynchronization period is defined as the number of coded bits between the beginning of the codeword corrupted by an error and the end of the codeword on which word resynchronization takes place. (MUC or terminating codewords associated with the first and last codeword in the resynchronization period are

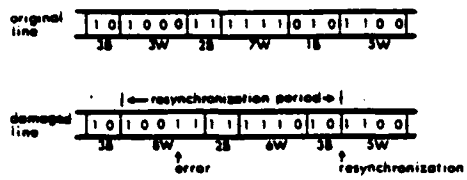


Fig. 4. Effect of an error on coded data.

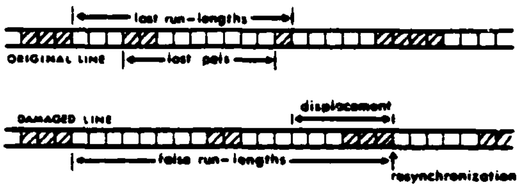


Fig. 5. Effect of an error on a scan line.

included.) Occasionally, the error does not cause loss of synchronization as the length of the corrupted codeword decoded by the decoder is the same as that formed by the coder. In this case, the resynchronization period is equal to the length of the corrupted codeword.

Fig. 5 shows the corresponding effect of the same error on the actual decoded scan line. A sequence of correct ("lost") run lengths is replaced by a sequence of incorrect ("false") run lengths. If the correct run lengths generated after resynchronization could be replaced in their true positions, the actual number of damaged pels would be reduced to the sequence labelled "lost" pels. The amount by which these correct run lengths are shifted to the right or left of their true positions is called the displacement.

The resynchronization periods and displacements of the modified Huffman code, as described above, cannot be easily calculated and therefore were measured experimentally using computer simulation techniques. Single errors were randomly inserted in the coded data obtained from a document and a computer program was used to decode simultaneously both a correct version of the coded data stream and a version containing the single error. When a corrupted codeword was detected, its position was recorded and decoding was continued until resynchronization in terms of both "color," i.e., black and white, and codeword length took place between the 2 data streams. About 2000 errors were inserted into the data obtained from each coded document in such a way that once one error had been inserted, no further error was inserted until resynchronization had occurred.

Table VI summarizes the results obtained for 4 documents scanned at high resolution. The median values show that for half the errors, the number of lost runs and lost pels is less than 50 and 20 pels, respectively. These are equivalent to disturbances of 6.2 and 2.5 mm along a scan line, respectively. These results show that in most cases the modified Huffman code recovers very quickly from disturbances and indicate that one of the correlation processes described in Section V-D could be used to reduce substantially the damage caused by a large proportion of such errors.

C. End-of-Line and Fill Bit Errors

Three effects or events are considered.

1) Lost EOL: This occurs when an error corrupts the EOL codeword in such a way that it cannot be recognized. Using

TABLE VI
RESYNCHRONIZATION PERIODS FOR MODIFIED HUFFMAN CODE

DOCUMENT	RESYNC PERIOD CODED BIT		LOST RUNS (PELS)		LOST PELS (PELS)		DISPLACEMENT PERCENTAGE ≤ 5 PELS
	AVERAGE	MEDIAN	AVERAGE	MEDIAN	AVERAGE	MEDIAN	
1	26	18	391	54	215	21	28%
4	24	16	122	29	77	13	39%
5	24	17	217	54	133	22	29%
7	27	17	140	69	96	13	29%

any of the error concealment techniques described in Section V-E, two lines adjacent to the EOL will be replaced by a single line.

2) Premature EOL: This occurs when an error occurs in the fill bits. It creates a spurious or false EOL and an extra line may be inserted in the document. Premature EOL's may be recognized since they produce coded lines with fewer coded bits than specified by the MSLT.

3) False EOL: An error corrupts a coded scan line in such a way as to create a spurious EOL. An extra line will be added to the document.

Statistically, the average number (A) of EOL's hit by single random errors during the transmission of a document is equal to the number of errors occurring on the document multiplied by the proportion of coded bits assigned to EOL codewords

$$A = (E \times N) \times \left(\frac{L \times M}{N}\right) = ELM \tag{6}$$

where E is the error rate, N is the total number of coded bits, L the length of the EOL codeword and M the number of scan lines. Thus A is independent of the number of bits required to transmit the document and depends only on the error rate. For example, if E is 1 in 10^5 and M is 2376 lines, then A is equal to 2.8. If bursts of errors are considered, each of which is spread over B bits, then the above equation becomes $A = E(L + B - 1)M$. If, for example, E is a burst error rate of 1 in 10^5 and B is 10 bits then A equals 0.5. On an actual telephone line, the burst length can vary widely and the values of A may differ from those given in this simple analysis.

Computer simulation methods were used to assess the frequency of occurrence of the three events listed above. Four of the CCITT documents scanned at high resolution were coded with a MSLT of 20 ms and the coded data was subjected to the first of the 4 real error patterns recorded by the University of Hannover. The corrupted data was decoded by a computer program and a record of the three EOL error events was kept. Three test runs were performed for each document. In the first run, the beginning of the error pattern coincided with the start of the message (Fig. 3). In runs 2 and 3, the start of the coded data coincided respectively with the 1024th and 2048th bits of the coded data. The results obtained for the 3 runs are given in Table VII.

The real error pattern has an average bit error rate of 7.3 in 10^4, a burst error rate of 7.1 in 10^5 (guard length of 100 bits), 99 percent of bursts have a spread of between 9 and 16 bits, the average burst spread is 10 bits and the probability of a block of 2048 bits being in error is 0.08. However, these statistics disguise the fact that the distribution of the errors in this real error pattern is very uneven. Because of this and

TABLE VII
EFFECTS OF ERRORS ON EOL SEQUENCES FOR ONE-DIMENSIONAL CODING

DOCUMENT	1			4			5			7		
TEST RUN	1	2	3	1	2	3	1	2	3	1	2	3
LOST EOLS	1	3	3	5	6	0	12	9	13	5	9	5
PREMATURE EOLS	3	4	2	5	6	2	3	3	7	9	9	5
FALSE EOLS	0	1	0	1	0	1	0	0	1	0	0	1
DAMAGED LINES	24	26	24	59	60	55	83	78	81	70	75	75
NO OF ERROR BURSTS	33			87			79			66		

because the number of bits required to code each document varies, the values in Table VII differ considerably. For example, the average burst error rate for document 5 is higher than that for the other documents and this produces a large number of lost EOL's.

The error rates quoted above for the real error pattern are regarded as being rather high. For example, measurements by Balkovic *et al.* [20] indicate that 95 percent of connections on the Bell Telecommunications Network have lower error rates and 50 percent of connections have burst error rates of less than 1 in 10^6 for a transmission rate of 4800 bit/s. Although error rates for many telephone networks are not available (error rates are difficult to classify because of their wide diversity), there is no evidence to show that, in general, they differ substantially from Balkovic's results. Thus it may be concluded that the error disturbance on most documents will be small and substantially lower than indicated by the results in Table VII.

D. Error Concealment Techniques

Unless the number of decoded pels between 2 successive EOL codewords is equal to 1728, then it can be assumed that an error has occurred in a transmitted line. In this case one of the following error concealment processes may be adopted:

1) replace the damaged line by an all white line;
2) repeat the previous line;
3) print the damaged line;
4) use a line-to-line processing or correlation technique to reconstruct as much of the line as possible.

The run lengths decoded after resynchronization are displaced from their correct positions on the scan line. If the displacement is more than about 4 pels, then the picture information on the damaged line will become disassociated from that on adjacent lines. Since this effect is very noticeable and causes a disturbing streak across a page, it is usually preferable to use method 1) or 2) above rather than to print the damaged line. For small displacements, however, method 3) may provide a simple means of minimizing the loss of information due to an error.

Correlation methods 4) take advantage of the fact that the recovery period is often short and attempt to retain as much as possible of the correctly decoded data on a damaged line. This is achieved by attempting to locate the damaged run lengths. One method [21] is to measure the correlation between groups of pels on the damaged line with corresponding groups on the adjacent lines above and below. Where the correlation is good, generally at the beginning and end of the damaged scan line, the scan line data is used to reconstruct the line. The part of the scan line which is assumed to be damaged is then replaced by a corresponding part of the previous line. Other interesting reconstruction methods are described in the literature [3], [22].

VI. Selection of a Two-Dimensional Coding Scheme

There is considerable interest in two-dimensional facsimile coding techniques, particularly in Japan, as is illustrated by Yasuda in this issue [32]. It is asserted that such coding schemes speed up the transmission of documents, especially when they are scanned at high resolution, without significantly increasing system costs. Although two-dimensional coding schemes are more vulnerable to transmission errors since the effect of a single disturbance can propagate over several lines it is felt that the increase in document degradation is not large enough, in general, to deter their use. As a result, Japan [23] proposed that a two-dimensional code called the relative element address designate (READ) code should be included as an option in the T.4 Recommendations for Group 3 equipment. This code combines some of the techniques used in two earlier coding schemes, RAC and EDIC, (these and the READ code are described by Yasuda), and is specifically designed to be an extension of the Group 3 one-dimensional coding scheme. Interest in this code is substantial since it has proved to be more efficient than RAC, EDIC, and many other two-dimensional coding proposals.

SGXIV subsequently received further contributions concerning two-dimensional coding schemes which fall into 2 categories. IBM Europe [24], the 3M Company [25], AT&T [26], and the British Post Office [27] proposed schemes which are also designed to be direct extensions of the one-dimensional code. The Federal Republic of Germany [28] and the Xerox Corporation [29] proposed schemes based on predictive coding [30]. SGXIV agreed a procedure for measuring the compression efficiencies and for assessing the error susceptibilities of the codes and during 1979 the codes were extensively tested by SGXIV delegates. The performances of the codes were then examined at a SGXIV meeting in Kyoto, Japan, in November 1979. A comparison of the codes shows that, from a technical point of view, there is little difference between them in terms of their compression efficiency and error susceptibility. The READ code was strongly supported because it had already been built into a large number of commercial machines. However, some SGXIV delegates suggested that a number of modifications to the READ coding procedure would simplify its implementation without significantly changing its compression efficiency. The following alterations were suggested:

1) Vertical mode coding should be restricted so that the examination of the reference line does not extend beyond ±3 pels. The statistics for the coding elements obtained for the READ code show that horizontal mode coding was nearly always more efficient than vertical mode coding when the examination of reference line is extended beyond 3 pels. This restriction simplifies the implementation since it is not necessary to code every changing element by both horizontal and vertical mode coding.

2) The necessity to add insertion bits (bit stuffing) into the coded data should be avoided. It was generally agreed that the use of insertion bits in the READ code to ensure a unique line

synchronization sequence would add extra implementation complexity.

3) The EOL codeword should be made the same as that used in the one-dimensional coding procedure. This ensures that the code retains its resynchronization properties and avoids the need for bit stuffing.

4) The code should cater for future extensions. In particular, a number of delegates expressed their desire for the two-dimensional code to provide an uncompressed mode. Later it may also be desirable to include more sophisticated coding procedures such as feature extraction or pattern recognition techniques and the coding of gray or colored areas.

After considerable discussion a suitable compromise, called the modified READ code, was proposed by the Japanese delegation. This proposal incorporated many of the features described above and was readily supported by SGXIV.

VII. The Two-Dimensional Coding Scheme

The modified READ code is a line-by-line scheme in which the position of each changing element on the coding line is coded with respect to either the position of a corresponding changing element on the reference line, which lies immediately above the coding line, or with respect to the preceding changing element on the coding line. After the coding line has been coded, it becomes the reference line for the next coding line. In order to prevent the vertical propagation of damage caused by transmission errors, no more than $K - 1$ successive lines are two-dimensionally coded and the next line is one-dimensionally coded. Usually K is set equal to 2 at normal resolution and set equal to 4 at high resolution. Before describing the coding procedure, it is necessary to define the changing pels and the 3 coding modes used in the coding procedure.

A. Definition of Changing Picture Elements

Definition: A changing picture element is an element whose "color" (black or white) is different from that of the previous element along the same line.

The coding algorithm makes use of 5 changing elements situated on the coding and reference lines. These are defined below with examples given in Fig. 6.

- a_0: The reference or starting changing element on the coding line. Its position is defined by the previous coding mode as described in Section VII-C. At the start of the coding line, a_0 is set on an imaginary white changing element situated just before the first actual element on the coding line.
- a_1: The next changing element to the right of a_0 on the coding line. This has the opposite color to a_0 and is the next changing element to be coded.
- a_2: The next changing element to the right of a_1 on the coding line.
- b_1: The next changing element on the reference line to the right of a_0 and having the same color as a_1.
- b_2: The next changing element on the reference line to the right of b_1.

If any of the coding elements a_1, a_2, b_1, b_2 are not detected at any time during the coding of the line, then they are set on an imaginary element positioned just after the last actual element on the respective scan line.

B. Definition of Coding Modes

The coding procedures uses 3 coding modes which are defined below and illustrated by the examples given in Fig. 6.

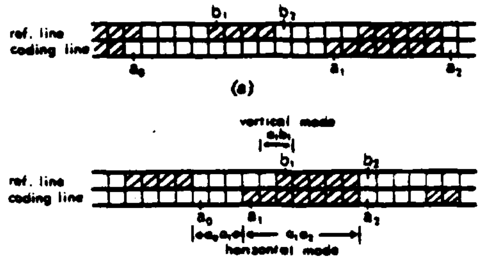


Fig. 6. (a) Pass mode. (b) Vertical and horizontal modes.

1) Pass Mode Coding: This is identified when the position of b_2 lies to the left of a_1. The purpose of the pass mode is to identify white or black runs on the reference line which are not adjacent to corresponding white or black runs on the coding line. The pass mode is represented by a single codeword in the two-dimensional code table (Table VIII).

2) Vertical Mode Coding: When this mode is identified, the position of a_1 is coded relative to the position of b_1. The relative distance a_1b_1 can take on one of seven values $V(0)$, $V_R(1)$, $V_R(2)$, $V_R(3)$, $V_L(1)$, $V_L(2)$, and $V_L(3)$ each of which is represented by a separate codeword. The subscripts R and L indicate that a_1 is to the right or left, respectively of b_1, and the number in brackets indicates the value of the distance a_1b_1.

3) Horizontal Mode Coding: If vertical mode coding cannot be used to code the position of a_1, then its position must be coded by horizontal mode coding. That is, the run lengths a_0a_1 and a_1a_2 are coded using the codewords $H + M(a_0a_1) + M(a_1a_2)$. H is the flag codeword "001" taken from the two-dimensional code table (Table VIII) and $M(a_0a_1)$ and $M(a_1a_2)$ are codewords taken from the appropriate modified Huffman code tables to represent the colors and values of the run lengths a_0a_1 and a_1a_2.

C. The Coding Procedure

Having determined the next set of changing elements a_1, a_2, b_1, and b_2, the coding procedure identifies the next coding mode, selects the appropriate codeword from Table VIII and then resets the reference element a_0 as defined below. The coding procedure is formally defined by the flow diagram shown in Fig. 7 and basically consists of 2 steps.

a) Step 1:

i) If b_2 is detected before a_1 then a pass mode has been identified and the code word "0001" is issued. The reference element a_0 is set on the element below b_2 in preparation for the next coding.

ii) If a pass mode is not detected, proceed to Step 2.

b) Step 2: Determine the number of elements which separate a_1 and b_1.

i) If $|a_1b_1| \leq 3$ then code the relative distance a_1b_1 by vertical mode coding. Set a_0 on the position of a_1 in preparation for the next coding.

ii) If $|a_1b_1| > 3$ then code the positions of a_1 and a_2 by horizontal mode coding, i.e., transmit the codewords $H + M(a_0a_1) + M(a_1a_2)$. After the coding, a_2 is regarded as the new position of the reference element a_0.

It is possible to vary the above procedure without affecting the compatibility between coder and decoder but further studies into the use of these variations are required. For example, it is possible to restrict the use of the pass mode to a single

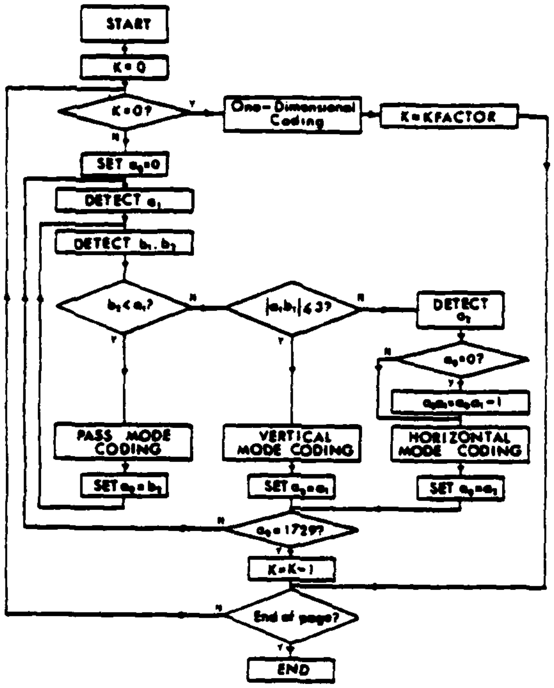


Fig. 7. Flow diagram for two-dimensional code.

pass made to prevent long sequences of pass modes which might give inefficient coding. Also, if $|a_1b_1| \leq 3$, each a_1 may be coded by both vertical and horizontal mode coding and the most efficient coding mode chosen as in the original READ code. However preliminary tests have not indicated that these particular variations lead to increased compression factors.

D. Coding the First and Last Elements on a Line

If horizontal mode coding is used to code the first element on the coding line, then the value of a_0a_1 is replaced by $a_0a_1 - 1$ to ensure that the correct run-length value is transmitted. Therefore, if the first element on a line is black, then the first codeword $M(a_0a_1)$ will be that which represents a white run of zero length.

The coding of the line continues until the imaginary changing element situated just after the last actual element on the coding line has been coded. Thus exactly 1728 elements are coded on each line and the receiver can check that each line has been correctly decoded.

E. The Code Table

The two-dimensional code table is given in Table VIII and is also drawn in the form of a code tree in Fig. 8. The code tree is constructed so that it contains the codeword "0000000" which is then extended to form the EOL codeword 11 X "0" + "1". The remaining codewords are then added to the code tree according to the relative frequencies of the required coding elements. These frequencies were obtained by computer simulation tests on the CCITT documents. Finally the two-dimensional extension codeword (Section VII-I) is assigned to the shortest remaining codeword. This construction method ensures that the same unique EOL codeword is used whether a line is coded by the one- or two-dimensional coding procedure.

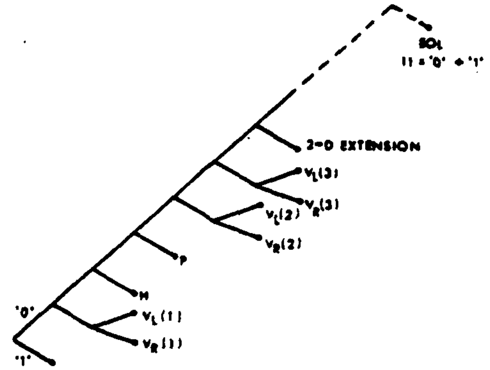


Fig. 8. Code tree for modified READ code.

TABLE VIII
THE TWO-DIMENSIONAL CODE TABLE

MODE	ELEMENTS TO BE CODED		NOTATION	CODEWORD
PASS	b_1, b_2		P	0001
HORIZONTAL	a_0a_1, a_1a_2		H	001 + $M(a_0a_1)$ + $M(a_1a_2)$
VERTICAL	a_1 JUST UNDER b_1	$a_1b_1 = 0$	V(0)	1
	a_1 TO THE RIGHT OF b_1	$a_1b_1 = 1$	$V_R(1)$	011
		$a_1b_1 = 2$	$V_R(2)$	000011
		$a_1b_1 = 3$	$V_R(3)$	0000011
	a_1 TO THE LEFT OF b_1	$a_1b_1 = 1$	$V_L(1)$	010
		$a_1b_1 = 2$	$V_L(2)$	000010
		$a_1b_1 = 3$	$V_L(3)$	0000010
2-D EXTENSIONS 1-D EXTENSIONS				0000001XXX 000000001XXX
END-OF-LINE CODEWORD (EOL)				000000000001
1-D CODING OF NEXT LINE 2-D CODING OF NEXT LINE				EOL + '1' EOL + '0'

$M(a_0a_1)$ and $M(a_1a_2)$ are codewords taken from the modified Huffman code tables given in Tables 1 and 2. The bit assignment for the XXX bits is 111 for uncompressed mode.

F. EOL codeword, Tag Bits, Fill Bits, and Return to Control

Each EOL codeword is followed by a single tag bit, a "1" or a "0", which indicates that the next line is one or two dimensionally coded, respectively. "Fill" bits consisting of variable length strings of "0"'s are inserted, when required, at the end of a coded line and before the EOL and tag bit. The return to control (RTC) signal consists of 6 consecutive EOL codewords, each of which is followed by a "1" tag bit.

G. K Factor

As stated earlier, it is recommended that, after a one dimensionally coded line, not more than $K - 1$ successive lines are two dimensionally coded, where K is equal to 2 for documents scanned at normal resolution, and 4 for those scanned at high resolution. More scan lines than suggested by the value of K can be one dimensionally coded without affecting compatibil-

ity if this proves useful in terms of either compression or error susceptibility.

H. Uncompressed Mode

Both one- and two-dimensional coding of some detailed documents leads to localized data expansion where the number of bits exceeds the number of pels. Documents containing screened photographic images, or areas of cross hatching as on some business forms can produce this effect. To cater for these situations, an uncompressed mode, proposed by IBM [31], has been suggested as an option to the two-dimensional coding scheme. Entry to the uncompressed mode on a one- and two-dimensionally coded line is achieved by using the one- and two-dimensional extension codewords, respectively, given in Table VIII, with the bits XXX set to 111. The other combinations of XXX are reserved for other as yet unspecified extensions. However, on a one dimensionally coded line, the coder does not enter uncompressed mode following a codeword ending in the bit sequence "000". This prevents the detection of false EOL's by the decoder caused by concatenation of this sequence with the one-dimensional extension codeword.

The following example of an uncompressed mode is given in Recommendation T.4. Once the uncompressed mode has been entered, the source data itself is transmitted, with a "0" representing a white pel and a "1" representing a black pel. Each group of 5 successive "0"'s must be followed by an insertion bit "1". The insertion bits are discarded by the decoder. The corresponding code table is

Image Pattern	*Codeword*
1	1
01	01
001	001
0001	0001
00001	00001
00000	000001

The uncompressed mode "1" insertion process allows the use of the following 5 exit codewords:

Codeword	*Image Pattern*
0000001T	none
00000001T	0
000000001T	00
0000000001T	000
00000000001T	0000

The flag bit T denotes the color of the next run: black is 1, white is 0.

At the present time, the uncompressed mode is still under review by SGXIV and has not been fully tested. Procedures are being considered which will determine the optimum entry and exit points for the uncompressed mode.

VIII. Performance of the Two-Dimensional Coding Scheme

Table IX lists the number of coded bits for the 8 CCITT documents using the two-dimensional coding procedure. At normal (low) resolution with $K = 2$ at a transmission rate of 4800 bit/s, the average transmission times are 47.3 and 54.1 s with MSLT equal to 0 and 20 ms, respectively. At high resolution with $K = 4$, the corresponding average times are 74.0 and 90.6 s, respectively. These 4 average times are, respectively, 16.4, 10.6, 33.3 and 25.1 percent lower than the corresponding average times achieved using one-dimensional coding. At high resolution, K = infinity and MSLT set to 0 ms, the average transmission time is 61.5 s which is 44.7 percent lower than the transmission time for one-dimensional coding.

Table X lists the frequencies of the coding elements obtained for 4 of the CCITT documents scanned at high resolution with $K = 4$.

Useful increases in the compression factors can be obtained by coding each line by both one- and two-dimensional coding and selecting the coded line with the fewest bits. This selection is carried out under the restriction that no more than $K - 1$ successive lines are two dimensionally coded. For documents 1, 4, 5, and 7 scanned at high resolution (MSLT = 0 ms, $K = 4$), the number of coded bits obtained were 194622,

TABLE IX
Number of Coded Bits Obtained for Two-Dimensional Group 3 Coding

DOCUMENT	LOW RESOLUTION		HIGH RESOLUTION		
	0 msecs K = 2	20 msecs K = 2	0 msecs K = 4	20 msecs K = 4	0 msecs K = INFINITY
1	130684	189093	207660	337697	175704
2	106851	144442	157163	260721	117304
3	207584	237407	326297	399332	260527
4	408261	440175	654436	726964	585074
5	226285	255384	353172	427584	288655
6	150572	181221	225879	304069	164085
7	402333	420119	651643	690479	585135
8	184369	210457	264029	333066	183674
AVERAGE	227117	259787	355034	434989	295019

TABLE X
Frequencies of Coding Elements, 7.7 Lines/mm, $K = 4$

DOC	P	H	V(0)	$V_L(1)$	$V_L(2)$	$V_L(3)$	$V_R(1)$	$V_R(2)$	$V_R(3)$
1	1664	2654	16445	5853	979	316	5947	1035	267
4	7730	10032	59075	22003	4190	921	24790	4671	902
5	3224	4310	35528	11396	1592	362	10850	1386	262
7	8612	12392	49861	21878	4301	1770	15859	2822	327

TABLE XI
Effects on Errors on EOL Sequences for Two-Dimensional Coding

DOCUMENT	1			4			5			7		
TEST RUN	1	2	3	1	2	3	1	2	3	1	2	3
LOST EOLS	2	5	7	10	4	7	3	5	3	3	3	5
PREMATURE EOLS	4	3	8	6	6	6	5	6	9	2	2	2
FALSE EOLS	0	1	1	1	0	1	0	1	0	0	0	0
DAMAGED LINES	49	61	42	113	114	106	64	65	97	127	119	121
NO OF ERROR BURSTS	30			81			42			51		

611132, 333837, and 597175, respectively. These values are on average 7 percent lower than those for the corresponding values obtained with a fixed value of K.

Table XI lists the results obtained by subjecting the real error pattern to the coded data obtained for 4 of the documents scanned at high resolution (MSLT = 20 ms, $K = 4$). Three test runs were carried out as described in Section V-C.

VIII. Conclusion

The one-dimensional run-length coding scheme and the two-dimensional modified READ code which we have described in this paper are expected to form the basis of a CCITT standard (Recommendation T.4) which will allow digital facsimile equipment to interwork on national and international general switched telephone networks.

The evaluation of the basic one-dimensional code, particularly in respect to its compression efficiency, susceptibility to errors and implementation complexity, indicates that it provides a good 1 min facsimile standard. The optional two-dimensional code provides larger reductions in transmission times for high resolution documents and is considered to be one of the most efficient of the two-dimensional codes proposed so far. The error susceptibility of the codes is difficult to assess both objectively and subjectively. However, computer simulation tests and manufacturers' field experience indicate that the error disturbance on most documents transmitted using the one-dimensional code is small. The two-dimensional code is potentially more susceptible to errors, and only time will tell whether it will always provide acceptable copy quality on national and international telephone networks.

The formation of the Group 3 standard, as in the case of the standardization of Group 2 analog facsimile equipment, has dramatically increased the degree compatibility between facsimile machines and has opened up many new applications for this communication method. Although the recent steady increase in the installation of Group 2 equipment is likely to continue, it is expected that the Recommendation T.4 will stimulate the growth of digital facsimile equipment. Currently efforts are being made to achieve compatibility between communicating text handling terminals (e.g., Teletex) for use on either telephone or data networks. Future international standards will also aim for the transmission of facsimile over data networks and for compatibility between facsimile and text transmission systems.

Acknowledgment

The authors acknowledge the efforts of innumerable individuals and organizations throughout the world who have supported the CCITT activities leading to the draft Recommendation T.4. They also acknowledge many useful discussions with A. T. Bence and R. E. F. Clint made during the preparation of this paper, and express their thanks to the Director of Research of the British Post Office for permission to publish this paper.

References

[1] H. G. Musmann and D. Preuss, "Comparison of redundancy reducing codes for facsimile transmission of documents," *IEEE Trans. Commun.*, vol. COM-25, pp. 1425–1433, Nov. 1977.

[2] T. S. Huang, "Coding of two tone images," *IEEE Trans. Commun.*, vol. COM-25, pp. 1406–1424, Nov. 1977.

[3] U. Rothgordt, G. Aaron, and G. Renelt, "One-dimensional coding of black and white facsimile pictures," *Acta Electron.*, vol. 21, no. 1, pp. 21–37, 1978.

[4] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: Univ of Illinois Press, 1949.

[5] J. Capon, "A probabilistic model for run-length coding of pictures," *IRE Trans. Inform. Theory*, vol. IT-5, pp. 157–163, 1959.

[6] R. B. Arps, "The statistical dependence of run-lengths in printed matter," *Codierung, Nachrichtentech. Fachberichte*, vol. 40, pp. 218–226, 1971.

[7] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, Sept. 1972.

[8] H. O. Burton and D. D. Sullivan, "Errors and error control," *Proc. IEEE*, vol. 60, pp. 1293–1301, Nov. 1972.

[9] R. E. F. Clint, "Burst error protection for digital facsimile using a microprocessor," presented at the 1979 Picture Coding Symp., Ipswich, England, July 9–11.

[10] H. Meyr, H. G. Rosdolsky, and T. S. Huang, "Optimum run-length codes," *IEEE Trans. Commun.*, vol. COM-22, pp. 826–835, June 1974.

[11] M. Takagi and T. Tsuda, "A highly efficient run-length coding scheme for facsimile transmission," *Electron. Commun. Jap.*, vol. 58A, no. 2, pp. 30–38, 1975.

[12] T. S. Huang and A. B. Hussain, "Facsimile coding by skipping white," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1452–1460, Dec. 1975.

[13] A. J. Frank, "Uniform decoding of minimum-redundancy codes," U.S. Patent 3 883 847, May 13, 1975.

[14] D. C. Van Voorhis, "Variable-length to fixed-length conversion of minimum-redundancy codes," U.S. Patent 4 044 347, Aug. 23, 1977.

[15] U. Rothgordt and G. Renelt, "Intermediate ternary code: A redundancy reducing run-length code for digital facsimile," *Electron. Lett.*, vol. 13, pp. 749–750, 1977.

[16] R. E. F. Clint, "A comparison of two coding techniques suitable for digital facsimile transmission," British Post Office Telecommunication Res. Dep. Rep. 727, Oct. 1978.

[17] EIA and BFICC, "Contribution to question 2/XIV point B1," CCITT SGXIV Doc. No. 3, Mar. 1977.

[18] E. N. Gilbert and E. F. Moore, "Variable length binary encodings," *Bell Syst. Tech. J.*, pp. 939–966, July 1959.

[19] B. Rudner, "Construction of minimum-redundancy codes with an optimum synchronization property," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 478–487, July 1971.

[20] M. D. Balkovic, H. W. Klancer, S. W. Klare, and W. G. McGruther, "High-speed voiceband data transmission performance on the switched telecommunications network," *Bell Syst. Tech. J.*, vol. 50, pp. 1349–1384, Apr. 1971.

[21] U. Rothgordt, "Localization of disturbances in digitally transmitted facsimile pictures," presented at the 1979 Picture Coding Symp., Ipswich, England, July 9–11.

[22] Y. Eto, Y. Hirano, and T. Fukinuki, "Facsimile signal decoding which localizes the effect of run-length code errors," *IEEE Trans. Commun.*, vol. COM-25, pp. 1170–1173, Oct. 1977.

[23] Japan, "Proposal for draft recommendation of two-dimensional coding scheme," CCITT SGXIV Doc. No. 42, Aug. 1978.

[24] IBM Europe, "Proposal for two-dimensional coding scheme," CCITT SGXIV Doc. No. 64, Jan. 1979.

[25] 3M Company, "Results of studies on previously proposed two-dimensional coding schemes," CCITT SGXIV Doc. No. 74, Mar. 1979, corrigendum, July 1979.

[26] AT&T, "Proposal for two-dimensional facsimile coding scheme," CCITT SGXIV Doc. No. 81, Mar. 1979.

[27] British Post Office, "Proposal for optional two-dimensional coding scheme for Group 3 facsimile apparatus," CCITT SGXIV Doc. No. 77, Mar. 1979.

[28] Federal Republic of Germany, "Two-dimensional coding scheme," CCITT SGXIV Doc. No. 82, Mar. 1979.

[29] Xerox Corporation, "Proposal for an optional two-dimensional coding system for Group 3 apparatus," CCITT SGXIV Doc. No. 84, Apr. 1979.

[30] D. Preuss, "Two-dimensional facsimile coding using a Markov model," *Nachrichtentech. Z.*, vol. 28, H.10, S, pp. 358–363, 1975.

[31] IBM Europe, "Uncompressed mode enhancement to IBM coding scheme," CCITT SGXIV Doc. No. 80, Mar. 1979.

[32] Y. Yasuda, "Overview of digital facsimile coding techniques in Japan," this issue, pp. 830–845.

AD-A117 026 DELTA INFORMATION SYSTEMS INC JENKINTOWN PA F/G 14/3
GRAPHIC/SYMBOL SEGMENTATION FOR GROUP 4 FACSIMILE SYSTEMS,(U)
APR 82 A R DEUTERMANN DCA100-81-C-0024

UNCLASSIFIED NCS-TIB-82-2 NL

2 OF 2

AD A
117026

END
DATE
FILMED
8-82
DTIC

Appendix B

Combined Symbol Matching Facsimile Data Compression System

APPENDIX B

Combined Symbol Matching Facsimile Data Compression System

WILLIAM K. PRATT, SENIOR MEMBER, IEEE, PATRICE J. CAPITANT, MEMBER, IEEE, WEN-HSIUNG CHEN, ERIC R. HAMILTON, MEMBER, IEEE, AND ROBERT H. WALLIS

Abstract—A facsimile data compression system, called combined symbol matching (CSM), is presented. The system operates in two modes: facsimile and symbol recognition. In the facsimile mode, a symbol blocking operator isolates document symbols such as alphanumeric characters and other recurring binary patterns. The first symbol encountered is placed in a library, and as each new symbol is detected, it is compared with each entry of the library. If the comparison is within a tolerance, the library identification code is transmitted along with the symbol location coordinates. Otherwise, the new symbol is placed in the library and its binary pattern is transmitted. Nonisolated symbols are left behind as a residue, and are coded by a two-dimensional run-length coding method. In the symbol recognition mode, the library is prerecorded and each entry is labeled with its ASCII code. As each character is recognized, only the ASCII code is transmitted.

Computer simulation results are presented for the CCITT standard documents. With text-predominate documents, the compression ratio obtained with the CSM algorithm in the facsimile mode exceeds that obtained with the best run-length coding techniques by a factor of two or more, and is comparable for graphics-predominate documents. In the symbol recognition mode, compression ratios of 250:1 have been achieved on business letter documents.

Manuscript received September 25, 1979; revised November 9, 1979.

The authors are with Compression Labs, Inc., 10440 North Tantau Avenue, Cupertino, CA 95014.

0018-9219/80/0700-0786$00.75

I. Introduction

MOST facsimile coding systems previously developed have been based on the concept of run-length coding [1]. Run-length coding methods provide a relatively high compression ratio for a graphics type of document or an alphanumeric document containing a small amount of text [2]. But, the achievable compression ratio drops appreciably if a document is filled densely with alphanumeric characters because the black and white run-lengths become quite short. Dense alphanumeric documents can be efficiently coded by symbol recognition techniques in which individual symbols are detected and coded by a prototype library code [3], [4]. However, such a method cannot effectively handle documents containing a mixture of alphanumerics and graphics. One proposed approach to this problem has been to segment a document into strips containing alphanumeric text or graphics data, and then code the former by symbol matching and the latter by run-length coding [5]. The problems with this approach are the difficulty of document segmentation and the drop in compression performance if the segmentation is not accurate. This paper introduces a new concept of hybrid symbol-matching/run-length coding in which a document is dynamically segmented into symbol and graphics regions [6].

Conceptually, the symbol versus graphics segmentation process employed in the facsimile compressor is quite simple. A document is scanned line by line, and all isolated symbols that are expected to recur in the document are extracted and coded by a symbol-matching process. The remainder of the document, called the residue, is coded by two-dimensional run-length coding. This segmentation method permits document symbols to be coded by symbol matching without interference from the graphics portions of a document, and eliminates symbols from that portion of the document which is run-length coded. The result is an efficient match between the type of data and the chosen coding methods.

The symbol-matching process previously described has been adapted to recognize alphanumeric characters in a document. In this symbol recognition mode of operation, the document is represented by conventional printer codes: character, space, carriage return, etc.

The following sections describe the combined symbol matching (CSM) algorithm for both the facsimile and symbol recognition modes of operation.

II. Facsimile Coding Mode

The block diagram of Fig. 1 describes the basic elements of the CSM coding system for facsimile coding. In operation, a number of scan lines equal to about two to four times the average character height are stored in a scrolled buffer. This data is then examined line by line to determine if a black pixel exists. If the entire line contains no black pixel, this information is encoded by an end-of-line code. If a black pixel exists, a blocking process is conducted to isolate the symbol. For those isolated symbols, further processing is required to determine if a replica of the symbol under examination already exists in the library. This process involves the extraction of a set of features, a screening operation to reject unpromising candidates, and finally a series of template matches. The first blocked character and its feature vector are always put into the prototype library, and as each new blocked character is encountered, it is compared with each entry of the library that passes the screening test. If the comparison is successful, the library identification (ID) code is transmitted

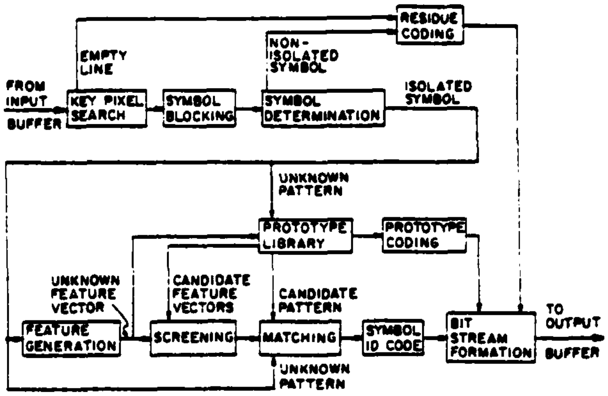


Fig. 1. CSM facsimile coder.

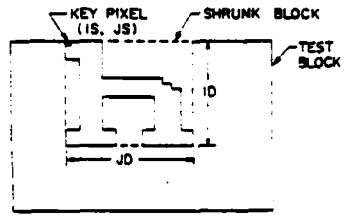


Fig. 2. Block shrinking.

along with the location coordinates of the symbol. If the comparison is unsuccessful, the new symbol is both transmitted and placed in the library. Those areas of a document in which the blocker cannot isolate a valid symbol are assigned to a residue, and a two-dimensional run-length coding technique is used to code the residue data. The following sections describe key elements of the coder in greater detail.

A. Symbol Blocking

The function of the symbol blocker is to examine the input buffer in a systematic fashion, and to locate the position and size of any isolated characters. Fig. 2 illustrates the blocking process. A black pixel in the buffer, denoted by the character "1" is considered to be a *key pixel* whenever the four neighbors located above it and to its left are white, as shown below

000

01.

Whenever a key pixel is encountered, the blocker is initiated. The blocker extracts those pixels from the buffer that are contiguous with the key pixel, or enclosed by a set of contiguous black pixels. For example, with the lower case letter "e," all black pixels and the enclosed white "island" will be extracted by the blocker.

B. Feature Extraction

The most straightforward method to determine whether a match exists between an unknown symbol and one of the symbols stored in the library is to perform a template match between the unknown and every library symbol. However, a two-dimensional template match is costly in terms of processing time and equipment. A method of reducing the number of such matches is required. The approach that has been taken is

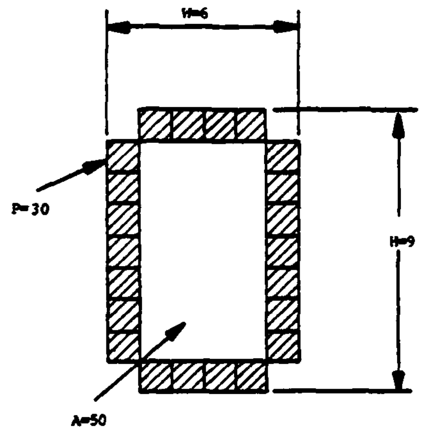


Fig. 3. Perimeter features. W = symbol width, M = symbol height, P = symbol outer perimeter, A = black area plus enclosed white area.

to extract a set of scalar "features" from the various symbols in the library. These features are used to reduce or "screen" the number of candidates for a template match to a tiny fraction of all the possibilities in the library.

The features used in the screening process are the block height, block width, symbol perimeter, and pixel area enclosed by the outer boundary of the symbol. Fig. 3 provides an example of features derived from a character.

C. Candidate Screening

The purpose of the screening process is to reduce the burden on the template matcher by passing only "good prospects" to the matcher. This is accomplished by calculating the feature space distance between the unknown and each library entry, and selecting the library candidate with the smallest distance as the best prospect for a match. If this match is rejected, the next best candidate is considered, and so forth. The distance "metric" used to determine how "close" an unknown is to a particular candidate is the "city block" distance defined by

$$D(U, C) = \sum_{I=1}^{N_F} |F_C(I) - F_U(I)| \qquad (1)$$

where $F_C(I)$ is the Ith feature of the candidate, $F_U(I)$ is the Ith feature of the unknown, $|\cdot|$ denotes the absolute value, $D(U, C)$ is the distance between the unknown and candidate, and N_F is the number of features.

D. Template Matcher

The template matcher forms a comparison between the binary patterns of a detected symbol and a library prototype symbol. Consider a two-dimensional binary pattern represented by $A(C, R)$ where $C = 1, 2, \cdots, N_C$ and $R = 1, 2, \cdots, N_R$. A conventional template matcher calculates the similarity between a pair of vector patterns $A(C, R)$ and $B(C, R)$ by summing the number of picture elements (pixels) for which $A(C, R)$ and $B(C, R)$ differ. This EXCLUSIVE OR error is defined as

$$E = \sum_{C=1}^{N_C} \sum_{R=1}^{N_R} A(C, R) \oplus B(C, R) \qquad (2)$$

where $\oplus$ denotes the Boolean EXCLUSIVE OR operation.

A major shortcoming of the conventional template matcher described above is that it treats all errors alike regardless of where they occur spatially. An improved matcher, to be described, utilizes a "weighted EXCLUSIVE OR" error criterion that is based on the context in which the error occurs.

The motivation behind the weighted EXCLUSIVE OR count error criterion may be appreciated by examining the EXCLUSIVE OR error (denoted $A \oplus B$) in Figs. 4 and 5. Compare the EXCLUSIVE OR pattern for the "c" and "o" in Fig. 4 with the pattern for the pair of "e's" in Fig. 5. Note that the EXCLUSIVE OR error count for the pair "c" and "o" (count = 23) is actually *less* than that for the pair of "e's" (count = 29) implying that, by this error metric, "c" and "o" are "closer" than the pair of "e's" are to each other. However, the error pattern for the pair of "e's," which should be declared a match, is composed of *sparsely distributed* pixels, while the error pattern for the "o" and "c" shows a *dense node* of error pixels corresponding to the missing right segment of the "o." One way to quantify the density of such a "node" is to form a summation in which the "local density" of every black pixel is merely the sum of all the pixels in its 3 X 3 neighborhood if the pixel is 1, and 0 if the pixel is 0. The patterns above labeled "weighted XOR error" have been calculated in this manner. Note that by this criterion, the associated counts indicate that the pair "c" and "o" are more separated (count = 131) than are the pair of "e's" (count = 73).

In the template matcher, the weighted EXCLUSIVE OR error is computed for nine translation shifts of a pair of patterns corresponding to horizontal and vertical single pixel shifts of the patterns. The minimum error is then compared to a threshold in order to determine whether or not a match should be declared. The value of the threshold is a nonlinear function of the symbol's black count, and is obtained by an empirically determined look-up table.

E. Library Maintenance

A fixed size library is used in the CSM system. The first blocked character and its feature vector occupy the first library slot. The subsequent library slots are occupied by those blocked characters for which no match is found. In order to prevent the library from overflowing, a scoring system is employed to track the usefulness of the library elements. When the library is filled, the least used prototype

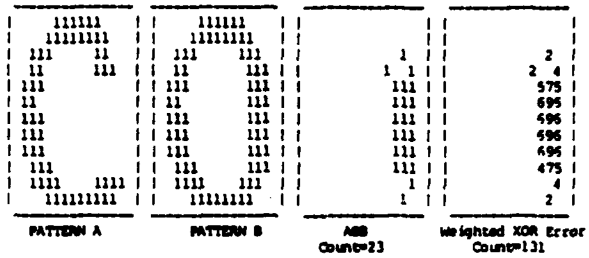


Fig. 4. Example of EXCLUSIVE OR pattern for *c* and *o*.

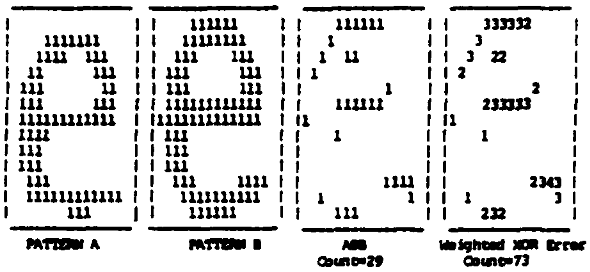


Fig. 5. Example of EXCLUSIVE OR pattern for pair of e's.

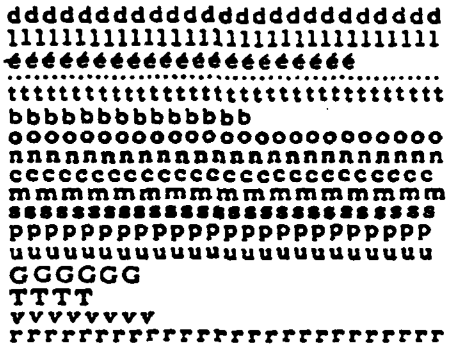


Fig. 6. Partial library plot of CCITT no. 4.

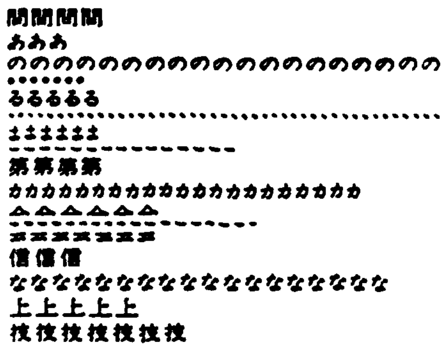


Fig. 7. Partial library plot of CCITT no. 7. (Symbol blocking performance using algorithm defined in [6].)

is bumped out of the library and replaced by the new prototype. At the receiver, the same size library and the same scoring system are utilized to maintain synchronization with the transmitter. With a library size of N elements, the scoring system gives every "new prototype" or "matched symbol" at least N chances for a match.

Figs. 6 and 7 contain partial library plots of isolated symbols from two facsimile documents, one a French journal article (CCITT #4), and the other a Japanese language document (CCITT #7). The first item on the list is the first isolated prototype symbol, and all following symbols represent matches to the prototype.

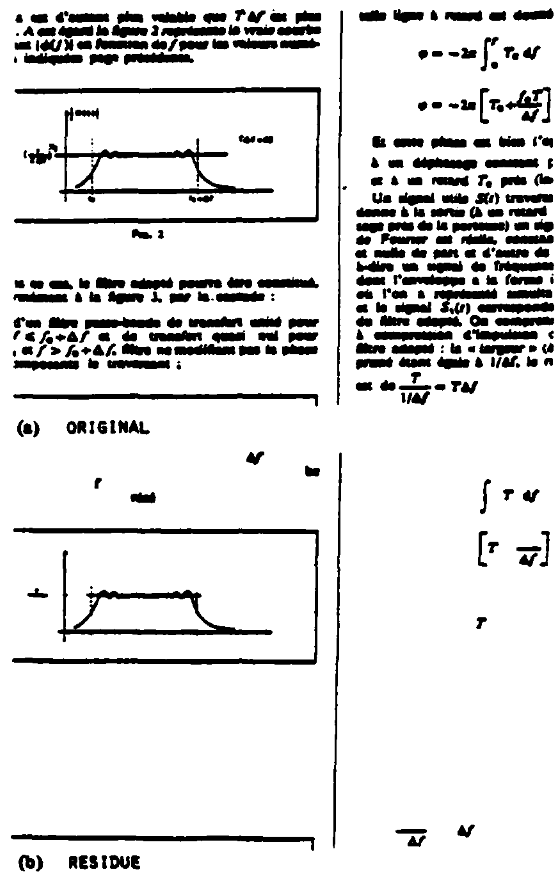


Fig. 8. Magnified section of CCITT no. 4 and its residue.

F. Prototype Coding

After a symbol has been blocked, a decision threshold is applied to each prototype element of the library that has passed the screening test. If a match is indicated, only the matching library ID and horizontal location with respect to the previous symbol are coded. Otherwise, the binary pattern of the blocked symbol is transmitted along with the symbol width, symbol height, and horizontal location, in addition to being placed in the library as a new prototype element.

The simplest method of prototype coding is to binary code the pixels within a block in a raster scan fashion. On the average, about 30 percent of the prototype code bits can be eliminated by scanning the prototype pixels in a folded "basket weave" sequence and applying one-dimensional Huffman coding of the run lengths. The disadvantages of this approach are additional implementation complexity and possible loss of bitstream synchronization when a channel error occurs. The binary coding approach has been adopted for a high-performance version of the CSM facsimile coder, and the folded run-length coding method is used for a very-high-performance version.

G. Residue Coding

In many documents, there are black pixel patterns that do not meet the criteria of prototype characters. Examples include exceptionally large or exceptionally small alphanumeric characters, segments of company logos, and segments of handwritten script. In the CSM system, these patterns are rejected by the symbol blocker, and then left behind as a residue to be coded by two-dimensional run-length coding. Fig. 8 presents a blow up of a section of a facsimile document (CCITT #4) and its corresponding residue.

Conceptually, the CSM system could employ any type of run-length coding method for residue coding. The selection should be made on the basis of coding performance, tolerance to channel errors, implementation complexity, and compatibility with facsimile standards. Considering these factors, a modified version of the CCITT two-dimensional run-length coding algorithm has been selected for the residue coder. By inhibiting the symbol matching process, the CSM coder will automatically revert to a pure residue coder, which can be made exactly compatible with the CCITT standard.

H. Transmission Code

The CSM facsimile coding system produces an asynchronous code that is dependent upon the contents of the document to be coded. Table I contains a detailed specification of the code elements and Fig. 9 contains a state diagram defining the code. The code words lengths in this specification have been optimized for a scan resolution of 8 × 8 pixels/mm.

TABLE I
CLI COMBINED SYMBOL MATCHING FACSIMILE CODE

CODE NAME	CODE DEFINITION	WORD SIZE (BITS)	DESCRIPTION
LINSYN	LINE SYNC	9	BEGINNING OF HORIZONTAL SCAN LINE, TRANSMITTED EVERY K LINES
SYMFLG	SYMBOL FLAG	1	0 = NO SYMBOL ON LINE 1 = AT LEAST ONE SYMBOL ON LINE
COLADD	COLUMN ADDRESS	11	HORIZONTAL LOCATION OF FIRST PIXEL ON LINE
MATFLG	MATCH FLAG	1	0 = NO MATCH 1 = MATCH
LIBID	LIBRARY INDEX	7	BINARY CODE OF LIBRARY INDEX
ROWPOS	ROW POSITION	2	VERTICAL MATCH SHIFT 00 = NO SHIFT 01 = UP SHIFT 10 = DOWN SHIFT
BLKSIZ	BLOCK SIZE	10	BLOCK SIZE (HEIGHT, WIDTH)
BLKCOD	BLOCK CODE	VAR.	BINARY CODE OF BLOCK CONTENTS
DELCOL	DELTA COLUMN	6,17	HORIZONTAL DISTANCE FROM CURRENT BLOCK TO NEXT BLOCK (UNIQUE CODE WORD INDICATES LAST SYMBOL ON LINE)
RESCOD	RESIDUE CODE	VAR.	TWO-DIMENSIONAL RUN-LENGTH CODE
STFBIT	STUFF BITS	VAR.	BITS INSERTED IN TRANSMISSION CODE TO PREVENT FACSIMILE CODE FROM ASSUMING STATE OF LINE SYNCH CODE

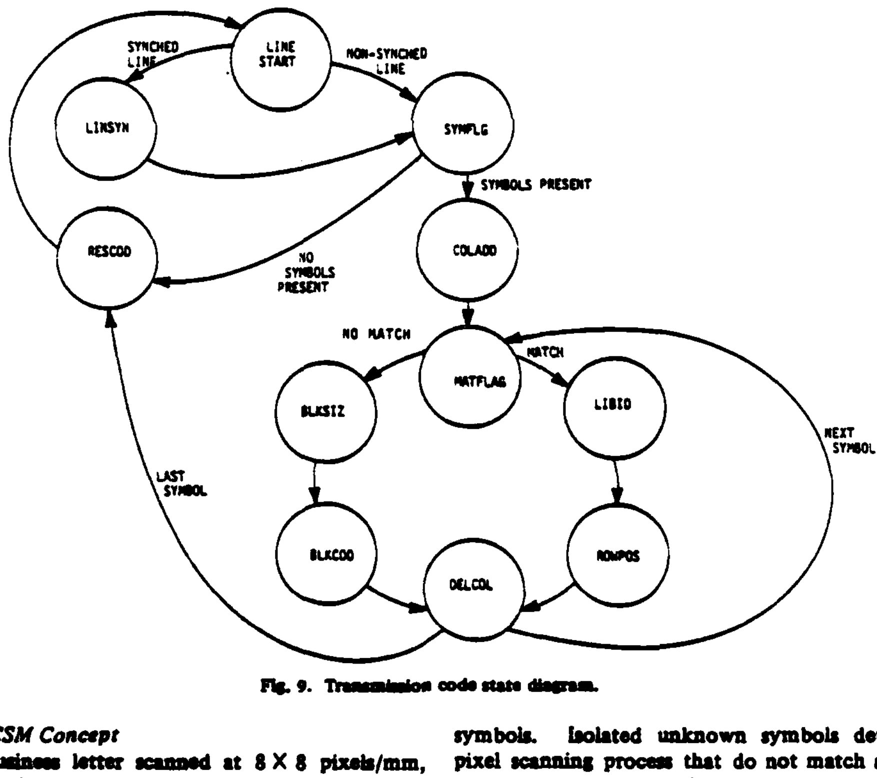


Fig. 9. Transmission code state diagram.

I. Extensions of CSM Concept

In a typical business letter scanned at 8×8 pixels/mm, about 40 percent of the compressed code bits are devoted to the transmission of prototype symbols. Almost all of this portion of the transmission code can be eliminated if the documents to be transmitted are restricted to a fixed set of symbols, for example, Courier typewriter font. In this case, the transmitter and receiver libraries can be prestored with symbols. Isolated unknown symbols detected in the key pixel scanning process that do not match a library entry can be placed in the residue for subsequent run-length coding.

The symbol matching process in the CSM system is not exact; a match tolerance is permitted between symbols to accomodate perturbations in symbol shape caused by the scanning process. As a consequence, in the basic CSM system, a reconstructed document is not an exact pixel-by-pixel replica

of the original document. Although symbol substitution errors are extremely rare, there may be applications in which exact coding is demanded. This mode of operation can be accomodated in the CSM system by a simple modification of the coder and decoder. At the coder, after a successful match, the EXCLUSIVE OR between the pair of matched symbols is formed and placed in the residue for subsequent run-length coding. At the decoder, the pixel arrays generated from reconstructed symbols and reconstructed residue are combined in an EXCLUSIVE OR fashion to correct for differences in the pair of matched symbols. In this manner, exact reproduction is achieved. However, the "overhead" associated with the exact reproduction mode of operation can reduce the achievable compression ratio by as much as 50 percent at 8 X 8 pixel/mm resolution.

III. Symbol Recognition Mode

The CSM algorithm achieves facsimile data compression by the matching of document symbols against a library of symbols accumulated during the document scan. If a match occurs, the library index is transmitted rather than the symbol binary pattern. This basic concept can be extended to perform symbol recognition by preloading the library with the binary symbol patterns of a predetermined set of symbol fonts. The coder can then operate in a symbol recognition mode in which only the ASCII codes are transmitted and all other document data such as a signature or logo are ignored.

A. *Line Tracking*

In the western world, printed matter is "read" from left to right and from top to bottom. Therefore, a symbol blocking system that transmits its output to a serial ASCII terminal must do the same. However, the CSM algorithm extracts characters from the document being scanned in a totally different fashion. As the line buffer scrolls through the page from top to bottom, the tallest of first encountered characters are removed from the document and processed through the recognition algorithm. Thus characters emerge from the CSM process in a sequence which would be totally incomprehensible if viewed in chronological sequence. In the conventional CSM facsimile transmission mode, this is of no consequence, since characters are placed in their appropriate address locations regardless of their order of occurrence. In the serial symbol recognition mode, the transmitter will assign each character an ASCII code, assemble the codes into lines, inserting blanks, line feeds, carriage returns, etc., and transmit the lines serially to the receiver. For single spaced or rotated documents, this "line tracking" is more difficult than one would imagine. The problem is basically that of grouping the characters into lines. Determining the sequence in which they should be transmitted is relatively easy since the characters may be sorted by their column addresses. A significant benefit of this serial ASCII mode is that no information on character location need be transmitted, since the correct sequence is all that is required in order to properly reconstruct the received document.

The line-tracking algorithm is based on a straight line fit of the key pixel coordinates of characters on a text line, as illustrated in Fig. 10. The straight line is defined parametrically as

$$R = S \cdot C + O \tag{3}$$

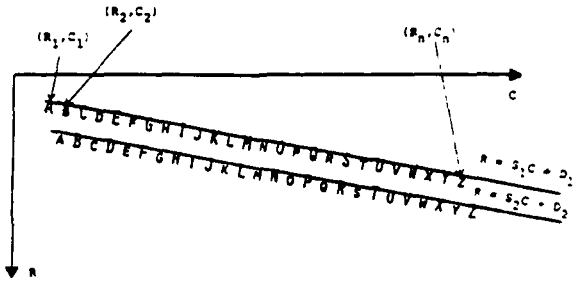


Fig. 10. Line tracking.

where R represents the row index, C is the column index, S denotes the text line slope, and O is its offset. As characters are encountered, they are assigned to the nearest straight line representing a text line. The algorithm is as follows:

1) The coordinates (C, R) of the first encountered character are used as a "seed" to start a cluster at $S = 0$, $O = R$.

2) The (C, R) coordinates of the next character encountered are used to compute $E = [R - S \cdot C]^2$ for the slope and offset of each cluster.

3) If the error is less than a threshold for a given cluster, the character is assigned to that cluster (next line). If it is greater than the threshold for all clusters, the oldest cluster is dumped, and a new cluster is started.

4) If the character was added to an existing cluster, the values of slope and offset are updated by use of minimum-mean-square error techniques.

B. *Handling of Special Characters*

A number of characters which consist of two "subcharacters" must be treated as special cases in the symbol-recognition mode. This is because the blocker/matcher would otherwise fragment them into their constituent parts and give misleading results. These characters are: (i), (j), (!), (?), (:), (;), (=), and ("). After recognition of the two parts of the character, the system will check if two compatible symbols are on top or almost on top of each other. If so, the two symbols are merged into one. For example two (·)'s on top of each other will be merged into a (:).

IV. Compression Ratio Evaluation

The CSM system has been extensively evaluated by computer simulation to optimize its performance and to determine its compression ratio with respect to other coding methods.

A. *Facsimile Mode Evaluation*

The CCITT document set of eight digitized documents of 200 X 200 line/in (8 X 8 pixels/mm) resolution, shown in Fig. 11, has been used for evaluation of the CSM system in its facsimile mode of operation. Tables II and III contain listings of the compression ratios for each of the documents for the high-performance and very-high-performance versions of the CSM algorithm, respectively. These tables also contain the bit allocations for each of the code elements defined in Table I.

Table IV presents a summary comparison of the compression ratios of the high-performance and very-high-performance CSM systems with several other facsimile coding methods. The modified Huffman code is the CCITT adopted standard for one-dimensional run-length coding [2]. The IBM code

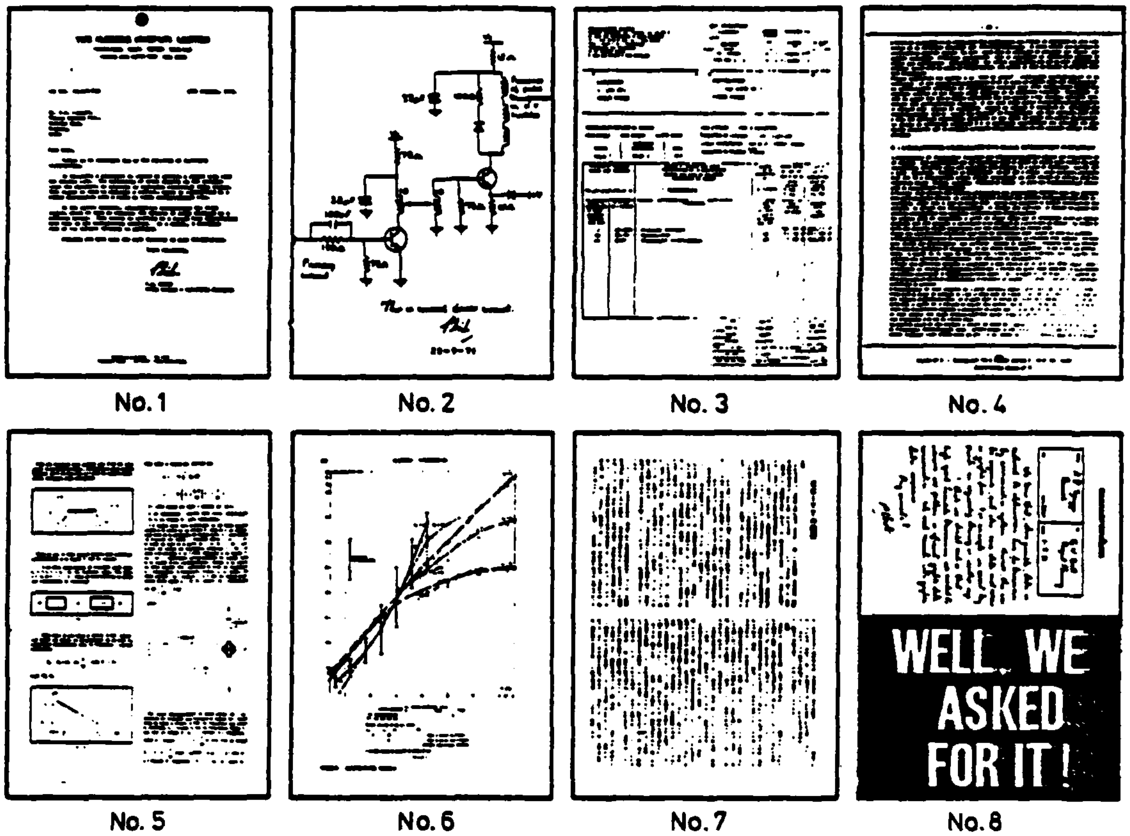


Fig. 11. CCITT facsimile document set.

[7], READ code [8], and BPO code [9] are proposals for a CCITT standard employing two-dimensional run-length coding. These algorithms all provide for an end-of-line code. All of the algorithms in Table IV have been simulated and evaluated on the same set of digitized documents scanned at the University of Hannover, Germany. The K factor indicates the number of lines in which the coder is operated in its two-dimensional mode before it reverts to a one-dimensional mode to limit the propagation of errors.

Comparison of the compression performance of these algorithms indicates that the CSM methods outperform the run-length coding techniques substantially for text-predominate documents, and perform at about the same level as the best of the two-dimensional run-length coding methods for graphics-predominate documents.

B. Symbol Recognition Mode Evaluation

The symbol recognition mode system has been tested with 86 sets of data, each containing 1000 samples of one of the 86 symbols of the Courier 10 font. In these tests, no mismatches occurred, and only very badly damaged characters were rejected.

Fig. 12 contains an example of a business letter and its reconstruction with the symbol matching coding mode of operation. It should be noted that the reconstructed letter has been printed with a different font than the original, however, the format and spacing of the two letters are in basic agreement. The compression factor obtained for this document for operation of the CSM system in the symbol matching mode is about 257 : 1 and for operation in the facsimile mode is about 49 : 1.

V. System Implementation

Although the CSM system is more complex to implement than a conventional two-dimensional run-length coding system, with the advent of high-speed and relatively inexpensive memory, discrete logic circuits, and microprocessors, implementation complexity has ceased to be a deterrent to the development of high-performance systems. A 100 X 100 lines/in (4 X 4 pixel/mm) facsimile coder using the CSM algorithm was introduced by Compression Labs, Inc. of Cupertino, CA, in Fall 1978. This unit utilizes a microprocessor to implement the algorithm for transmission at sub-minute page rates. A discrete logic implementation of the CSM algorithm is being developed by Compression Labs for transmission rates of less than 5 s for a 200 X 200 lines/in page.

VI. Summary

A new high-performance method of facsimile data compression, called CSM, has been introduced. The coding system involves segmentation of a document into symbols, that are coded by template matching, and into a residue of the remainder of the document, that is coded by two-dimensional run-length coding. Computer evaluation indicates that the compression factor for text-predominate documents is about

TABLE II
HIGH-PERFORMANCE CODER SUMMARY

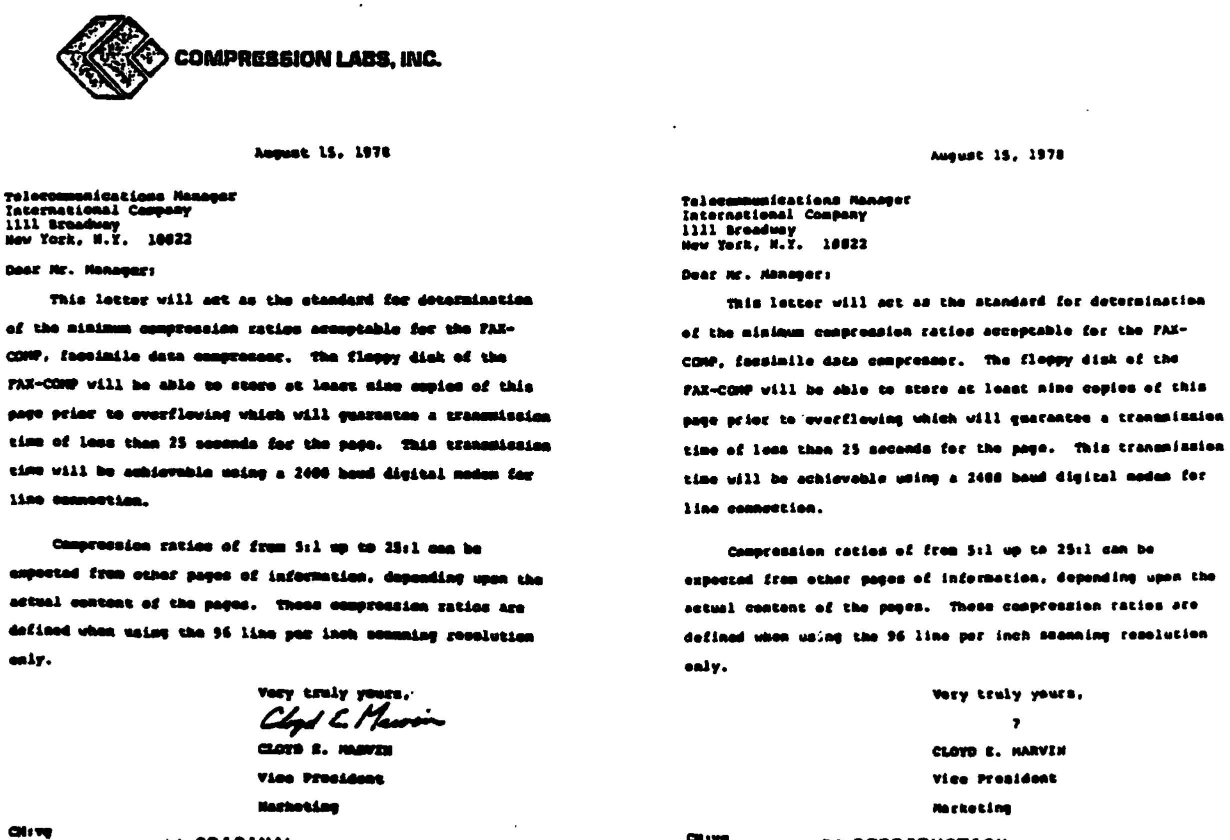
	DOCUMENT							
	1	2	3	4	5	6	7	8
STFBIT	699	387	2557	1613	1700	2299	6836	1792
LINSYN K=32	603	603	603	603	603	603	603	603
SYMFLG	2128	2128	2128	2128	2128	2128	2128	2128
COLADD	2497	385	4169	8767	4895	1980	11176	1584
MATFLG	988	37	1291	4015	1754	330	2522	156
LIBID	5985	168	7574	26292	10773	1253	9870	434
ROWPOS	1710	48	2164	7512	3078	358	2820	124
BLKSIZ	1330	130	2090	2590	2150	1510	11120	940
BLKCOD	24691	3404	47036	55360	44789	30534	406471	28115
DELCOL	8227	233	9341	34177	14055	2431	25692	1024
RESCOD	18594	91323	53880	20752	42014	93847	19845	181353
TOTAL	67452	98846	132833	163809	127939	137273	499083	218253
COMP. RATIO	54.5	37.2	27.7	22.4	28.7	26.8	7.4	16.8

TABLE III
VERY-HIGH-PERFORMANCE CODER SUMMARY

	DOCUMENT							
	1	2	3	4	5	6	7	8
STFBIT	439	266	1431	1182	1152	1560	3979	936
LINSYN K=32	603	603	603	603	603	603	603	603
SYMFLG	2128	2128	2128	2128	2128	2128	2128	2128
COLADD	2497	385	4169	8767	4895	1980	11176	1584
MATFLG	988	37	1291	4015	1754	330	2522	156
LIBID	5985	168	7574	26292	10773	1253	9870	434
ROWPOS	1710	48	2164	7512	3078	358	2820	124
BLKSIZ	1330	130	2090	2590	2150	1510	11120	940
BLKCOD	16841	2058	30929	37399	28513	20937	322780	18239
DELCOL	8227	233	9341	34177	14055	2431	25692	1024
RESCOD	18594	91323	53880	20752	42014	93847	19845	181353
TOTAL	59342	98846	115600	145417	111115	126937	412535	207521
COMP. RATIO	62.0	37.8	31.8	25.3	33.1	29.0	8.9	17.7

TABLE IV
COMPRESSION RATIOS FOR CODING OF CCLTT DOCUMENT SET WITH VARIOUS CODING ALGORITHMS

CCITT DOCUMENT	CCITT 1-D	READ K=4	READ K=32	IBM K=4	IBM K=32	RPO K=4	RPO K=32	CSM H.P. K=32	CSM V.H.P. K=32
1	15.2	21.8	24.9	20.6	23.2	20.6	23.2	54.5	62.0
2	15.1	28.7	38.0	24.1	29.1	25.7	32.6	37.2	37.8
3	8.7	13.6	16.3	13.2	15.6	13.3	15.8	27.7	31.8
4	5.3	6.6	7.1	6.6	7.2	6.5	7.0	22.4	25.3
5	8.5	12.7	14.7	12.3	14.1	12.4	14.3	28.7	33.1
6	10.2	19.0	25.1	17.6	22.5	18.1	23.5	26.8	29.0
7	4.8	6.1	6.7	6.1	6.6	6.1	6.6	7.4	8.9
8	7.9	15.1	20.2	12.9	15.8	14.0	18.2	16.8	17.7


COMPRESSION LABS, INC.

August 15, 1978

Telecommunications Manager
International Company
1111 Broadway
New York, N.Y. 10022

Dear Mr. Manager:

This letter will act as the standard for determination of the minimum compression ratios acceptable for the FAX-COMP, facsimile data compressor. The floppy disk of the FAX-COMP will be able to store at least nine copies of this page prior to overflowing which will guarantee a transmission time of less than 25 seconds for the page. This transmission time will be achievable using a 2400 baud digital modem for line connection.

Compression ratios of from 5:1 up to 25:1 can be expected from other pages of information, depending upon the actual content of the pages. These compression ratios are defined when using the 96 line per inch scanning resolution only.

Very truly yours,

Cloyd E. Marvin

CLOYD E. MARVIN
Vice President
Marketing

CM:vg

August 15, 1978

Telecommunications Manager
International Company
1111 Broadway
New York, N.Y. 10022

Dear Mr. Manager:

This letter will act as the standard for determination of the minimum compression ratios acceptable for the FAX-COMP, facsimile data compressor. The floppy disk of the FAX-COMP will be able to store at least nine copies of this page prior to overflowing which will guarantee a transmission time of less than 25 seconds for the page. This transmission time will be achievable using a 2400 baud digital modem for line connection.

Compression ratios of from 5:1 up to 25:1 can be expected from other pages of information, depending upon the actual content of the pages. These compression ratios are defined when using the 96 line per inch scanning resolution only.

Very truly yours,

CLOYD E. MARVIN
Vice President
Marketing

CM:vg

(a) ORIGINAL (b) REPRODUCTION

Fig. 12. Example of document compression in CSM mode.

twice that obtained with two-dimensional run-length coding and about the same for graphics-predominate documents.

The CSM system can be operated in a pure symbol recognition mode in which a document is coded by recognition of its alphanumeric symbols. Compression ratios greater than 250:1 can be achieved on business letters in this mode of operation.

References

[1] R. B. Arps, "Binary image compression," in *Image Transmission Techniques*, W. K. Pratt, Ed. New York: Academic Press, 1979.

[2] H. G. Musmann and D. Preuss, "Comparison of redundancy reducing codes for facsimile transmission of documents," *IEEE Trans. Commun.*, vol. COM-25, no. 11, pp. 1425-1433, Nov. 1977.

[3] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized text," *IEEE Trans. Comput.*, vol. C-23, pp. 1174-1179, Nov. 1974.

[4] W. K. Pratt, W. Chen, and C. Reader, "Block character coding," in *Proc. Soc. Photo-Optical Instrumentation Engineers*, vol. 66, pp. 222-228, Aug. 1976.

[5] W. Chen, J. L. Douglas, and R. D. Widergren, "Combined symbol matching—A new approach to facsimile data compression," in *Proc. Soc. Photo-Optical Instrumentation Engineers*, vol. 199, pp. 2-9, Aug. 1978.

[6] W. Chen, J. L. Douglas, W. K. Pratt, and R. H. Wallis, "A dual mode hybrid compressor for facsimile images," in *Proc. Soc. Photo-Optical Instrumentation Engineers*, Aug. 1979.

[7] J. L. Mitchell and G. Goertzel, "Two-dimensional facsimile coding scheme," in *Proc. Int. Communications Conf.*, pp. 8.7.1-8.7.5, 1979.

[8] "Proposal for draft recommendation of two-dimensional coding scheme," Rep. CCITT Study Group XIV, contribution no. 42, prepared by Japan.

[9] "Proposal for optional two-dimensional coding scheme for group 3 facsimile apparatus," Rep. CCITT Study Group XIV, contribution no. 77, prepared by British Post Office, Mar. 1979.

Appendix C

Proposal for Mixed Character Coded Text and Facsimile Areas on the Same Page

Questions : 16/VIII
10,11/XIV

TITLE : PROPOSAL FOR MIXED CHARACTER CODED TEXT AND FACSIMILE AREAS ON THE SAME PAGE.

SOURCE : MBLE S.A.
Belgium (Philips Subsidiary)

1. Summary

The aim of this contribution is to propose a method for :

a) the insertion of any number of rectangular facsimile areas, positioned anywhere throughout the page, within Teletex coded text;

b) the transmission of such mixed-mode pages in a manner that allows printing by relatively simple receivers.

2. General

There is an increasing desire for the transmission of pages presenting mixed text and images, on one hand. On the other hand, work has been undertaken to standardize Group 4 facsimile apparatus for transmissions over the public data networks. In this field these is a growing tendency towards the use of a common control procedure for Teletex and Facsimile communications.

In view of these considerations, the transport service and the end-to-end control procedures, already defined for Teletex in Recommendations S.70 and S.62 respectively, seem to offer valuable and flexible tools common to the three services (Teletex, Group 4 Facsimile, and Mixed test + image), owing to the offered advantages : universality and error recovery, a.o.

From this community of transport service and control procedures, two consequences arise :

- The existing family of facsimile equipments (including Group 3) make use of quite different procedures. To avoid making the new Group 4 equipments unduly costly, their direct communication possibilities with the existing range of facsimile equipments become highly questionable.
- Pure Teletex and pure Group 4 Facsimile services could perhaps be considered as simplified limit cases, derived from the more universal "Mixed mode" service.

Starting from this philosophy, the following family of terminals could then appear :

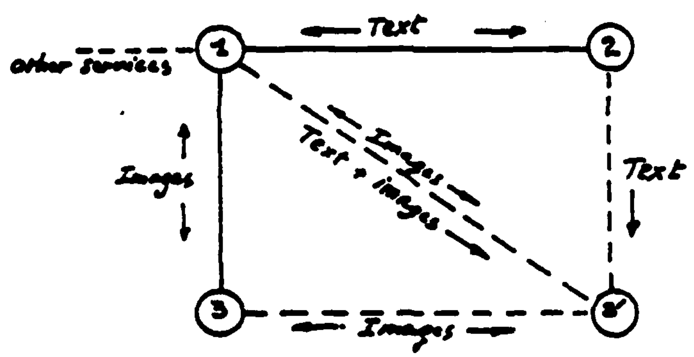
(1) Mixed mode terminal (includes : { Mixed text + image facility
Teletex
Group 4 Facsimile
... other services ...).

(2) Teletex terminal.

(3) Group 4 Facsimile terminal.

It is obvious that the simple machines (2) and (3) might be equipped with additional options, to obtain intermediate equipments with growing capabilities. As an example, by adding to the terminal (3) a character generator and some extra processing power, one could build a Group 4 terminal (3') having also the capability to receive and print character coded text.

The following communications over the public networks are then possible :



../...

3. Basic assumptions

In order to keep the benefit of previous standardization efforts in the Teletex and Facsimile fields, it seems reasonable to adopt following assumptions.

3.1. Coding

The encoding of the text characters, for transmission, shall be in accordance with the Teletex repertoire (Rec. S.61). The encoding of the facsimile information shall be in accordance with the unidimensional or bidimensional encoding schemes (Rec. T.4).

The encoding of some new commands needed by the mixed service shall also remain compatible with the mentioned encoding schemes.

3.2. Resolutions for facsimile and for text

Normally, terminals such as (1) and (3) (or 3'), capable of receiving ~~both~~ images ~~and text~~, will be equipped with printing mechanisms operating in accordance with the facsimile resolutions. Previous work done by others (see COM XIV-N°99) and by ourselves has shown that Teletex messages printed with T.4 facsimile resolutions are of good quality. Consequently :

- The horizontal and vertical resolutions for the facsimile parts of a page shall conform to Rec. T.4.
- The character pitch and the line spacings in the text parts shall conform to the basic Teletex standards given in Rec. S.61.

3.3. Shape of the image areas

For management simplicity, it is assumed that the image areas are rectangular-shaped, with edges parallel to the paper edges.

../...

4. The structure of a page containing mixed information

The composition and the transmission of a composite page needs a quantitative structuring of its area, compatible with both Teletex standards (character pitch and line spacing) and Facsimile resolutions. This can be obtained as follows.

4.1. Page partitioning

It is proposed to cover the page with a virtual grid of equally spaced vertical lines and equally spaced horizontal lines, thus defining horizontal (H) and vertical (V) coordinates.
The origin (0,0) should be ideally situated at the top left corner of the paper sheet.

The values of the H and V units will be determined in section 4.3.

4.2. Image positioning

Any image area shall be delimited by lines of the grid. This positioning method allows for the insertion of several images on a page, and does include the limit cases of :

- a slice of facsimile covering the total width of the page.
- a full facsimile page.

4.3. Character positioning

It is known (see also COM XIV-N°99) that the basic character pitch and lines spacings of Teletex do not correspond to an integer number of picture elements (horizontally) and scan lines (vertically) as defined in Rec. T.4 for facsimile.

When accepting a small approximation (less than 2 %) for the Teletex parameters, it is however possible to define a common grid as shown in fig. 1, where the horizontal and vertical units are given by :

H unit = 20 pels = $20 \times \frac{215}{1728} \approx 2.49$ mm (- 1.97 % w.r. to 2.54 mm)

V unit = 16 scan lines = $16 . \frac{1}{7.7} \approx 2.08$ mm (-1.83 % w.r. to $\frac{1}{12}$ inch).

It is to be noted that the negative approximations are in favour of the integrity of the reproduced pages.

With such a grid, any intersection (H,V) represents the virtual position of a text character, taking into account all basic line spacings defined in S.61.

5. Transmission of the mixed information

Depending on the objectives in view, the contents of a composite page may be segmented in various ways for transmission.

The "baseline by baseline" method proposed hereunder allows reception and printing of the mixed text and image information without the need of a full page storage. This is important when considering that the storage of a facsimile part needs about 20 times more memory capacity than the storage of a same area of Teletex. (One A4 Teletex page : 1.5...3 koctets, one A4 Facsimile : 30...60 koctets).

5.1. The "baseline by baseline" method

For better understanding, the method will be explained on basis of Fig. 2, which represents an example of an image associated with text in the same horizontal slice of the page.

Along each baseline of text, the first transmitted string contains in this case the Teletex coded characters of the left part; the next string contains the facsimile coded information corresponding to all the succesive facsimile lines (scanned from top to bottom) of the subimage slice comprised between the current base line and the previous one; the last string contains the Teletex coded characters of the right part.
Between strings of different nature, adequate delimiting commands must be inserted.

In the case of Fig. 2 the strings are as follows :

../...

Line 1 : <Text a> D_1 <EOL/Subimage a.../EOL> D_2 <Text a'/SVS/CR/LF>
(Teletex) (32 facsimile lines) (Teletex)

Line 2 : <Text b> D_1 <EOL/Subimage b.../EOL> D_2 <Text b'/SVS/CR/LF>
(48 facsimile lines)

⋮

Line 5 : <Text e> D_1 <EOL/Subimage e.../EOL> D_2 <Text e'/CR/LF>
(32 facsimile lines)

SVS stands symbolically for the Teletex parametric control function 'Select vertical spacing'.

D_1 is a delimiting command from Teletex to Facsimile fields.

D_2 is a delimiting command from Facsimile to Teletex fields.

It is seen that an image is split into several subimages which are transmitted with character strings inbetween.

It is the responsibility of the source system management to inject the delimiting commands and to possibly insert dummy characters (spaces e.g.) to maintain the correct alignement of the subimages.

5.2. <u>Definition of the "delimiting command"</u>

Two methods for indicating the delimiting points between facsimile and text coded information may be considered.

5.2.1. <u>Delimiting within the user data field of a CDUI command</u>

In this case the delimiting commands must be compatible with the repertoire of the current context.

- D_1 : "Text to image" command.
  It must be chosen in accordance with the rules applicable to Teletex control characters. It may be :

  ESC X (X to be defined in a new subrepertoire)
  or CSI + parameters.

The CSI command is preferred, since its parameter field may indicate, apart of the "delimiting" point, some extra characteristics of the next facsimile string, such as vertical resolution and number of pels per scan line in the subimage. The use of such possibilities is however mentioned as "for further study" in Rec. S.61, para. 3.3.4

- D_2 : "Image to text" command.

  It must be a string which is never encountered in the facsimile encoded data field. E.g. for unidimensional coding :

  $\cdot 001_H \cdot$<$00E_H$> where 001_H means EOL in hexadecimal

  or K.<EOL>

  The single EOL is maintained as normally to indicate the end of each scan line.

5.2.2. Delimiting at document procedure level

Another method consists in introducing a CDUI command at each delimiting point.

In S.62, the possibility is left open for adding parameters in the CDUI header. These parameters could define the delimiting command, along with additional parametric characteristics for the next user data field. In this case the need for defining specific "in text", D_1 and D_2 commands disappears.

This method however entails more overhead and thus less efficient communications.

5.3. Flow control

In basic Teletex, the error recovery mechanism is based on a RU (recovery unit) of one page; in facsimile and mixed mode, this implies a large memory capacity for storing at least one full page in the receiver.

../...

In Rec.62 Annex G, an optional error recovery mechanism is offered, allowing for RU of smaller size by introducing several recovery points within the page.

This last procedure could be a basis to implement also flow control in order to match the rate of information transfer to the capability of the receiver, without the need for inserting fill bits at the end of facsimile scan lines (method of Rec. T.4).

6. Conclusions

The proposed method for transmission of a page composed of mixed character coded areas and facsimile coded areas presents the following advantages:

- no need for a large storage capacity in the receiver
- high flexibility in the number and dimensions of images
- printing possible in terminals equipped with standard facsimile printing heads (T.4 resolutions)
- no need for the transmission of coordinates for images and text
- full compatibility with basic Teletex terminals
- possibility of changing the vertical facsimile resolutions for each image (and even for each subimage).

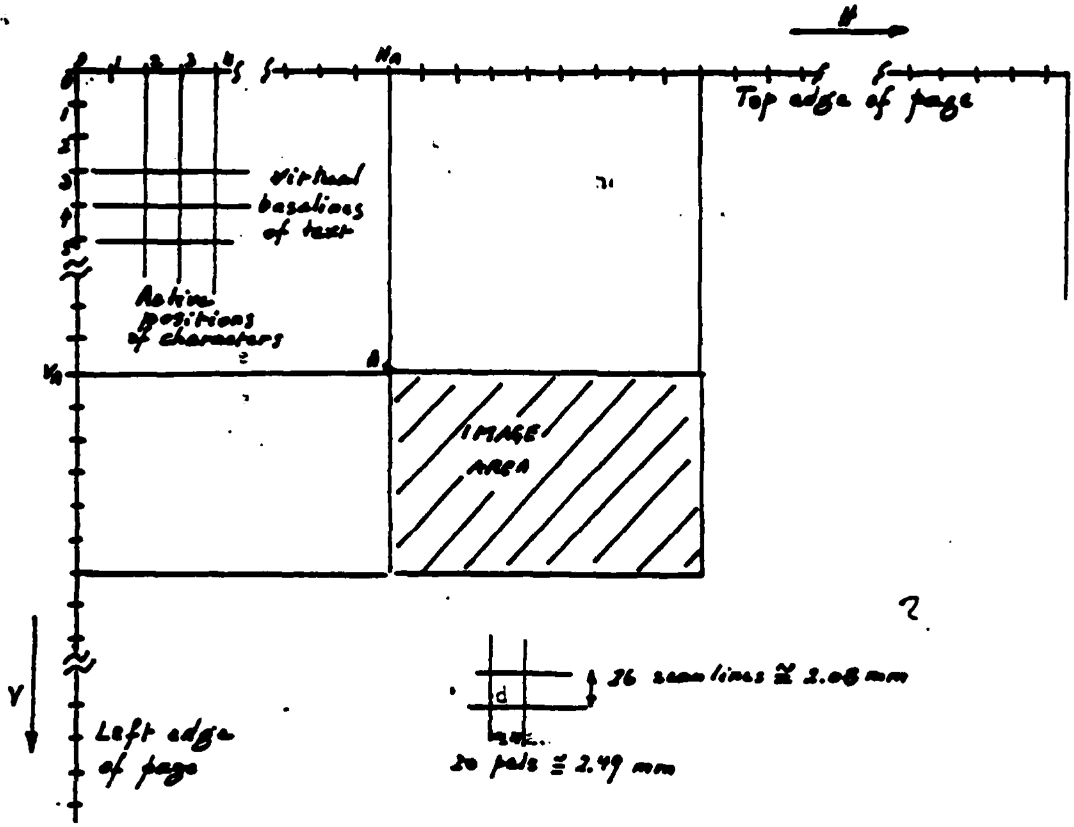


Fig 1: Page partitioning.

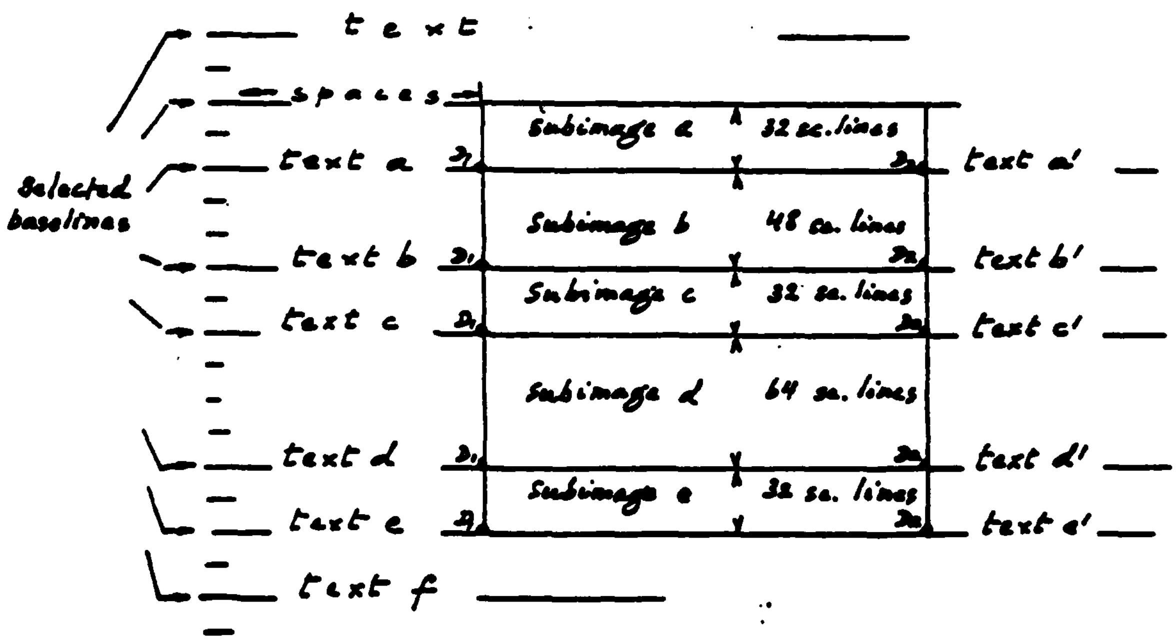


Fig. 2: Example of text+image in same page slice.